

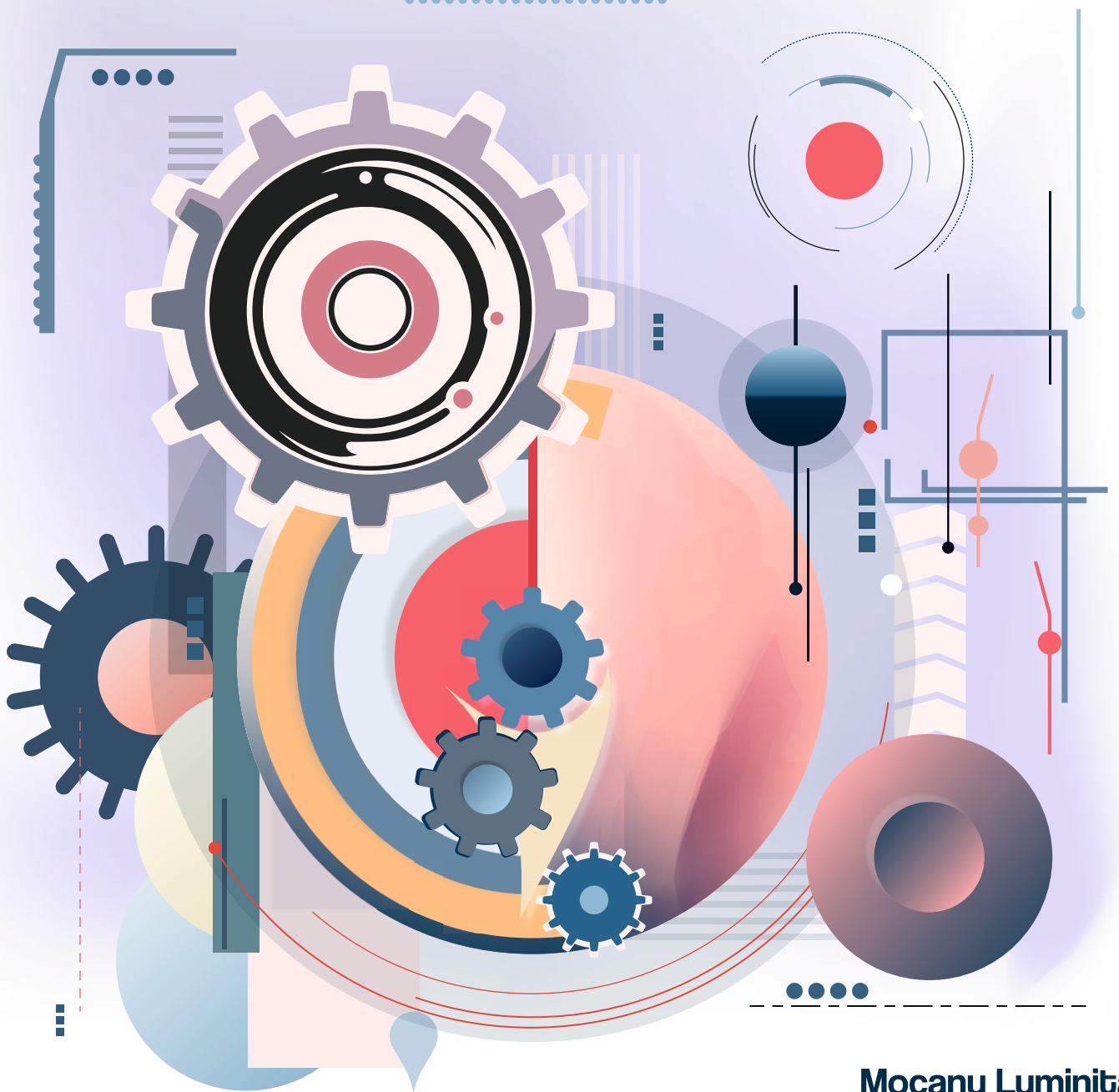


Erasmus+



Ghid metodic în predarea interdisciplinară a roboticii în educație

Methodological guide of
interdisciplinary teaching of robotics



Mocanu Luminița
Toncea Cristian
Petrescu Silvia
Toncea Angelica



Mocanu Luminița, Toncea Cristian,

Petrescu Silvia, Toncea Angelica

**Ghid metodic în predarea
interdisciplinară a roboticii în
educație
Methodological Guide of
interdisciplinary teaching of robotics**

ISBN 978-973-0-38956-2

Brăila/2023

REFERINȚĂ ȘTIINȚIFICĂ

Încă din 2016, Raportul World Economic Forum a creionat profesiile viitorului insistând asupra necesității de a asigura pe scară largă o educație cu o componentă majoră de tip STEM (Science, Technology, Engineering and Maths), termen ce se regăsește în peisajul educațional românesc sub numele de Robotică. Cercetările științifice ale ultimilor ani au demonstrat că robotica este o resursă importantă ce poate fi utilă în procesul de învățare pentru mai multe discipline din învățământul preuniversitar.

Proiectul ERASMUS+, „NEW TECHNOLOGIE, NEW WORLDS” și-a propus și a reușit să abordeze tematica roboticii în educație, propunând un set de metode și strategii de aplicare a roboticii educaționale în programa școlară. Pornind de la un glosar de termeni în domeniul roboticii și continuând cu literatura pe tematica robotică, proiectul a identificat ideile metodologice și pașii practici pentru realizarea unor lecții creative la orele de științe, limbi străine, tehnologia informației.

Potrivit recomandărilor ce se regăsesc și în documentele Comisiei Europene și ale OECD (Organizația pentru Cooperare și Dezvoltare Economică), Robotica reprezintă cea mai bună modalitate prin care elevii de astăzi se pot pregăti pentru profesiile viitorului ce vor fi bazate pe sisteme fizico-cibernetice avansate (roboți, drone, imprimante 3D). În același timp, Robotica poate reprezenta un important instrument menit să descopere și să rezolve principalele probleme legate de învățare (frustrare, neatenție), venind astfel atât în sprijinul elevilor, cât și al profesorilor.

Principala contribuție a acestui proiect este aceea că a reușit să identifice etapele principale pentru aplicarea Roboticii în Educație, fie pentru a implementa proiecte, fie pentru a preda Robotica cu alte discipline complementare sau nu.

Felicitări întregii echipe și mult succes în continuare!

Conf. univ. dr. Camelia Vechiu

Universitatea “Constantin Brâncoveanu”

Facultatea de Management Marketing în Afaceri Economice Brăila



GHID METODOLOGIC IN PREDAREA INTERDISCIPLINARA PE TEMATICA ROBOTICII IN EDUCATIE

CUVANT INAINTE

Odată cu progresul științei și tehnologiei au fost dezvoltate mai multe aspecte ale cercetării inovatoare care încearcă să descopere principalele probleme legate de învățare în rândul copiilor. Se știe că probleme precum frustrarea și neatenția, printre altele, afectează învățarea elevilor. În acest mod, robotica este o resursă importantă care poate fi folosită pentru a ajuta la rezolvarea acestor probleme, dând putere elevilor noștri pentru a-și îmbunătăți procesul învățării. În acest caz, instrumentele robotizate sunt utilizate în general luând în considerare două paradigme diferite: ca focus principal și ca focus secundar. De fapt, aceste paradigme definesc modul în care robotica educațională este implementată în școli. Majoritatea abordărilor l-au implementat ca accent principal, care este predarea roboticii. Cu toate acestea, există destul de multe lucrări care implementează robotica ca un accent secundar, care în prezent ajută procesul de învățare în mai multe discipline. Principala contribuție a acestei lucrări este o metodologie completă în civași pași pentru Robotica în Educație, fie pentru a ghida proiecte, fie pentru a o utiliza individual, fie pentru a preda robotica cu alte obiecte complementare sau nu. Experimentele noastre arată importanța elaborării unui plan de studiu și a unei metode de evaluare, deoarece procesul este iterativ și ar putea îmbunătăți rezultatele finale.

Ideea ghidului metodic a venit din participarea în calitate de beneficiar la un proiect Erasmus plus, 'NEW TECHNOLOGIES, NEW WORLDS' în care s-a abordat tematica roboticii în educație, programarea robotilor, literatura pe robotica, etica folosirii și recicalrii robotilor.

Ca noutate, ne-am extins lucrările anterioare propunând un nou set de metode cu linii directoare și strategii de aplicare a roboticii educaționale în programa școlară de la un glosar de termeni, literatura pe tematica robotică, clarificarea termenilor englezi la ora de limbă engleză, la programare la ora TIC, aplicarea la ora de științe, toate însoțite de un set de idei metodologice și pași practici pentru noi lecții creative. Unele planuri au fost create în limba engleză în ideea folosirii de către toți partenerii educaționali din proiectele Erasmus plus anterioare.

Vă propunem mai multe instrumente care au fost dezvoltate pentru a organiza temele de învățare a Roboticii pentru copii, inclusiv rezultatele dorite în timpul procesului de învățare. Un domeniu actual de cercetare care implică Educația și Tehnologia este să descopere și să rezolve principalele probleme legate de învățare, cum ar fi frustrarea și neatenția, printre altele. Cu progresele recente în știință și tehnologie, mai multe studii inovatoare au fost realizate în principal în lumina învățării asistate de tehnologie metode pentru a ajuta la rezolvarea acestor probleme. În această direcție, Robotica este un instrument important care a fost aplicat cu scopul de a împuternici elevii și de a le îmbunătăți învățarea, așa cum a fost deja verificat de mai mulți actori în educație. De fapt, educația este considerată o aplicație importantă în zona de Robotică, unde solicită elevii și profesorii să folosească instrumente destinate roboticii, astfel încât o mai bună înțelegere a ambelor este imperios necesară.

Coordonator,

PROF.LUMINITA MOCANU

With the advancement of science and technology, several aspects of innovative research have been developed that try to uncover the main problems related to learning among children. Issues such as frustration and inattention, among others, are known to affect student learning. In this way, robotics is an important resource that can be used to help solve these problems, empowering our students to improve their learning. In this case, robotic tools are generally used, considering two different paradigms: as a main focus and as a secondary focus. In fact, these paradigms define how educational robotics is implemented in schools. Most approaches have implemented it as their main focus, which is teaching robotics. However, there are quite a few works that implement robotics as a secondary focus, which currently helps the learning process in several disciplines. The main contribution of this paper is a complete methodology in several steps for Robotics in Education, either to guide projects, or to use it individually, or to teach robotics with other complementary objects or not. Our experiments show the importance of developing a study plan and evaluation method, as the process is iterative and could improve the final results.

The idea of the methodical guide came from participating as a beneficiary in an Erasmus plus project, 'NEW TECHNOLOGIES, NEW WORLDS', in which the topic of robotics in education, robot programming, robotics literature, the ethics of using and recycling robots were addressed.

As a novelty, we have extended our previous work by proposing a new set of methods with guidelines and strategies for applying educational robotics in the school curriculum from a glossary of terms, the literature on robotics, the clarification of English terms in the English class, at programming in ITC class, application in science class, all accompanied by a set of methodological ideas and practical steps for new creative lessons. Some plans were created in English with the idea of use by all educational partners from previous Erasmus plus projects who were invited at the online part of the dissemination.

We propose several tools that have been developed to organize Robotics learning tasks for children, including the desired outcomes during the learning process. A current area of research involving Education and Technology is to discover and solve the main problems related to learning, such as frustration and inattention, among others. With the recent advances in science and technology, several innovative studies have been conducted mainly in the light of technology-assisted learning methods to help solve these problems. In this direction, Robotics is an important tool that has been applied with the aim of empowering students and improving their learning, as it has already been verified by several actors in education. In fact, education is considered an important application area of Robotics, where it requires students and teachers to use tools dedicated to robotics, so a better understanding of both is imperative.

Coordinator, Luminita Mocanu

Robots for the English class

Topic

Robots

teacher:LUMINITA mOCANU

Aims

- To practise speaking skills
- To practise reading skills
- To review different verb tenses (past, present, future)

Age group

12 – adult

Level B1 / B2

Time 60 - 90 minutes

Materials

1. Robots student worksheet

2. Internet links:

- http://www.bbc.co.uk/search?v2=true&q=artificial_intelligence - BBC articles on artificial intelligence
- <http://news.bbc.co.uk/1/hi/programmes/hardtalk/2202825.stm> - BBC article and interview about the pros and cons of robots and AI.
- <http://www.robotshop.com/en/personal-domestic-robots.html> - Online shop with different domestic robots for sale (good for images / ideas for the final task)

Introduction

This lesson is based around robots and will encourage students to talk about how science is developing and influencing our everyday lives.

To introduce the topic there is a simple drawing dictation for students to do in pairs. The end product is a simple line drawing of a robot to introduce the topic. Task 1 gets students to think about what jobs they would like a domestic robot to do for them. As students have probably had most of their exposure to robots through films, Task 2 asks them to talk about the 'robot films' they have seen. Task 3 is a reading task for higher levels based around an article from the British Council's Culture Lab UK website. Task 4 encourages students to think about the past, present and future of everyday activities and Task 5 is an opportunity for students to create their own robots in teams and present their designs to the class. This task can be flexible depending on the age and level of the class.

Procedure

1. Introduce the topic – Robot back to the board

- Divide the class into two groups, A and B. Group A's will need a piece of paper and a pencil or pen and they need to be seated with their backs facing towards the board. It's vital they don't turn around and look at the board. Group B's should face the board and find a partner from group A and sit facing them. When the students are seated correctly, draw a simple line drawing of a robot on the board. Students in group B should describe the picture to their group A partner who should draw the same line drawing. 'Draw a rectangle at the top. In the rectangle draw a

small square near the top right hand corner and a small square in the top left hand corner. Under the two squares in the centre, draw a long thin rectangle. (this is the face of the robot!) Continue until the robot is completed. If the seating arrangements for this activity are impossible you can dictate the robot drawing to the whole group or give one of the students the picture and get them to do it.

2. Task 1 – Jobs for robots

This activity is to get students thinking about how a robot may help them in their every day lives in the future. Brainstorm ideas with the group (tidy my room, do homework, iron clothes, play computer games with me.....) then ask students to choose their top three. When all students have done this they can compare their answers and (depending on the level) defend their choices. You could remind students here of the 3 'd's – Robots tend to do work that is dirty, dangerous or dull.

3. Task 2 - Robofilms

I think we can assume that a lot of what people know about robots comes from films. This activity gives students a chance to talk about films they have seen with robots in and to discuss whether they believe any aspects of these films may one day become a reality.

4. Task 3 – Robot reading:

Jeeves Machine

This text is adapted from an article on a British Council website called Culture Lab UK. If your students are interested in science and technology you will find lots of interesting articles there. Depending on the level of your students you should consider pre-teaching some of the vocabulary beforehand. Students should read the article and then discuss the questions that follow in pairs or small groups.

5. Task 4 – Back to the future

This could be done as a group activity. You can also use it to revise the simple past or 'used to', present simple and simple future tenses with your group. Give some examples to get them on the right track. E.g.. Washed up by hand – put dishes in dishwasher – will have disposable dishes. You could think about areas like: Transport, technology, writing, social life, education, entertainment etc. Ask students to share their ideas at the end.

6. Task 6 – Design your own robot

Students work in small groups (design teams) to design their own robot. If you are teaching a very low level simplify the questions and get the students to draw the robot and think of the brand name and then simply write a list of what the robot can do. My robot is calledand it can..... Encourage students to use their imaginations and be as creative as possible. If they get into the idea and you have time, space and materials why not get them to build the prototype of their robot using cardboard boxes and household waste.

**RULES FOR
ROBOTS GET CLOSER**
Topic:Robots
teacher:LUMINITA MOCANU

Aims

- To practise speaking skills
- To practise reading skills
- To review different collocations

Age group

12 – adult

Level

B2-C1

Time

60 - 90 minutes

Materials

1. Robots student worksheet
2. Internet links: *rt.com*, *BBC News*, *The Guardian*

Warm-up

Do you think robots will be useful in the future or will they create problems? Why/why not?

Match the words on the left to their definitions on the right.

- | | |
|-----------------|---|
| 1. a politician | a. afraid |
| 2. rules | b. a piece of paper with important information on it |
| 3. behave | c. in every way |
| 4. exactly | d. sentences that say what you are and aren't allowed to do |
| 5. a document | e. someone who has a job in politics |
| 6. a designer | f. someone whose job is to decide how things should look |
| 7. scared | g. to act in a particular way |
| 8. manage | h. to control something |

Find the information

You are going to read an article about plans to make rules about how robots should behave in the future. Choose the best answer for each question, then scan the text on page three to check your answers.

1. What does the author of the document think we should be ready for?
a. robots that can feel
b. a robot revolution
c. a time that will make history
2. What will designers need to do when they make robots?
a. give lessons about how to use them
b. give them an identity number

c. make it possible to turn off the power

3. What could the European Union give people if robots take their jobs?

a. a small amount of money b. a bigger salary c. a place to live

4. What does one of the robot rules say?

a. a robot's owner is not be responsible for it

b. a robot can defend itself

c. a robot may hurt a human

5. What does the the writer of the document think will happen in the future?

a. robots will take all the jobs from humans

b. humans will learn to work together with robots

c. humans won't be able to live without robots

European politicians want rules for robots

1. European politicians want to make rules for how robots should behave in the future.

2. The rules in the document, which was written by Luxembourg MEP Mady Delvaux, may be used to say exactly what robots can and can't do.

3. In the document, the author writes that robots should have the legal name "electronic persons." There will be jobs that they have to do because of the rules, and if they cause accidents or break things, they will have to do something to fix it.

4. The author says she believes the world will soon see a robot revolution, and we should be ready for it.

5. The people that design the robots will have to make it possible for users to turn off the power. If not, it's possible that the robots could take control of themselves and cause problems.

6. At the same time, people should be able to use them and not be scared that robots will hurt them in any way.

7. The rules about robots will be for the makers, designers and users of the robots because they can't be made into machine code. Some of the rules say that a robot should not hurt a human, should follow orders, and should defend itself. But it can only do this if it doesn't break any of the other rules.

8. The document says that people could make a lot of money from robots. However, robots will also take some jobs, so countries might need to give everyone a small amount of money for living.

9. The document also says that in the future it is possible that robots will be more intelligent than humans, which could be dangerous.

10. So, people are asking questions about how safe humans will be if robot computer systems stop working, how robots should care for people, and how we can keep things private from robots.

11. For safety reasons, designers might have to register their robots, as well as give some code to people which they can use if their robots cause accidents. They may also have to get permission for new robots from a special group, who will decide if each robot will be good for humans.

12. The European Union could create a new agency to give advice to people about robot rules and how to manage their robots.

13. The author of the document believes that robots will have a place in the future, and that humans will work together with them.

14. She said: "A robot is not a human and will never be human."

Source: *rt.com, BBC News, The Guardian*

Checking understanding

Put 'T' (true) of 'F' (false) next to each statement below.

1. If robots cause any problems, they will have to fix their own mistakes.
2. Robots will be able to understand the rules.
3. Robots won't help people to become richer.
4. Robots may be smarter than humans in the future.
5. Humans are worried about how robots will look after them if they are sick.
6. Designers will not share code with the owners of their robots.

Find the words

Find a word or phrase in the text which means....

1. happening together (**phrase, P.6**):
2. instructions for a computer (**noun, P.7**):
3. put information on an official list (**verb, P.11**):
4. make a choice about something after thinking about your options (**verb, P.11**):
5. a business that gives people information and makes appointments (**noun, P.12**):

Collocations

Match the verbs on the left with the words on the right to make collocations from the text.

1. take a. advice
2. cause b. control
3. follow c. orders
4. break d. problems
5. give e. the rules

Talking point

Discuss any of the questions below.

1. Do you think it is a good idea to have rules for robots? Why/why not?
2. Do you think that robots will take a lot of jobs in the future? Why/why not?
3. Do you think that robots could become dangerous? Why/why not?

Key

1. Warm-up

Encourage a short discussion.

2. Key words

Students can work alone and then check in pairs.

1. e 2. d 3. g 4. c 5. b 6. f 7. a 8. h

3. Find the information

1. b 2. c 3. a 4. b 5. b

4. Checking understanding

1. T 2. F 3. F 4. T 5. T 6. F

5. Find the words

Students can work in pairs and then check with the teacher.

1. at the same time
2. machine code
3. register
4. decide
5. an agency

6. Collocations

Students can work in pairs and then check with the teacher.

1. b 2. d 3. c 4. e 5. a

7. Talking point

Monitor students conversations. Make a note of any typical errors and write up useful language on the board.

GUIDELINES FOR TEACHIING, 'I, ROBOT' BY ISAAC ASIMOV

teacher: Luminita Mocanu

I, Robot turns the world of science fiction literature on its head. Rather than telling the typical tale of a humanoid machine run amok (e.g., *Terminator*), SFWA Grand Master Isaac Asimov asks readers to imagine a world where robots protect us from our own worst nature. Beginning with a simple story about the relationship between a little girl and a limited-function robot, *I, Robot* moves on to explore, in subsequent stories, increasingly sophisticated thoughts, questions, and moral complexities. In the process, the book reveals Asimov's overarching vision of a future that entangles inextricably the humans and the machines.

The stories grew from Asimov's opinion that anyone smart enough to create robots would be smart enough to make sure that those robots wouldn't attack their makers. Conceived by Asimov as the Three Laws of Robotics—essential laws built into the robots' inner workings—these Laws freed science fiction writers to develop robots as characters instead of portraying them as monstrous things. *I, Robot* hints loudly that robots are a “better breed” than humans and though they were created to serve, they will inevitably become the masters.

At first Asimov had trouble getting “Robbie,” the opening story in *I, Robot*, published. But throughout the 1940s the subsequent tales appeared regularly in pulp science fiction magazines. In 1949, Asimov gathered the stories into a book that he wanted to call *Mind and Iron*. (The publisher prevailed with *I, Robot*.) The collection has enjoyed great success through the years and is offered by Spectra in a compact, affordable, and eminently readable edition for you—or your robot.

PLOT SUMMARY

“Introduction” 2057: Earth. An unnamed reporter for “Interplanetary Press” prepares to interview Susan Calvin, a seventy-five-year-old “Robopsychologist” who works for U.S. Robot and Mechanical Men, Inc., (U.S. Robots). Accused of being emotionless like a robot, Calvin argues that robots are more than mechanical parts, “They're a cleaner better breed than we [humans] are.” Calvin reminisces about early opposition to robots from labor unions (that were worried about competition) and religious groups (that were worried about sacrilege). Against these anti-robot arguments she holds the memory of an early robot model named Robbie, which was sold in 1996 as a nursemaid for a little girl. Calvin begins to tell the story. “Robbie”1998: Earth. Robbie the Robot plays hide-and-go-seek outdoors with his charge, nine-year-old Gloria Weston. Robbie lets her win. Gloria is a demanding but charming girl who loves Robbie. At Robbie's gestured urging (he cannot speak), she begins to tell him his favorite story, Cinderella, but her mother, Grace, calls them to come inside. Grace does not trust Robbie with her daughter and badgers her husband, George, to get rid of him until George finally gives in. Gloria is heartbroken when her parents take away Robbie. In an attempt to distract her, her parents decide to take Gloria on a trip to New York, hoping

the excitement of the city will take her mind off Robbie. While in New York, they tour the U.S. Robots factory. Gloria spies Robbie, one of “the robots creating more robots.” She runs toward him—right into the path of a moving tractor. Before anyone else has time to react, Robbie snatches Gloria out of harm’s way. Because Robbie has saved Gloria’s life, her mother grudgingly allows the robot to return to the family. At this point, the frame story (Susan Calvin talking to the reporter) resumes. Calvin tells us that robots were banned from Earth between 2003 and 2007. To ensure the company’s survival, U.S. Robots started developing mining models for other planets. Calvin recalls two troubleshooters, Mike Donovan and Gregory Powell, who worked with the experimental designs in “the teens.”

ABOUT THIS AUTHOR

Isaac Asimov (January 2, 1920–April 6, 1992)

Isaac Asimov was born in Russia but his family moved to New York when he was three years old. A self proclaimed “child prodigy,” he could read before first grade and had an almost perfect memory. Asimov credits his early intellectual development to public libraries: “My real education, the superstructure, the details, the true architecture, I got out of the public library. For an impoverished child . . . the library was the open door to wonder and achievement.” Asimov became fascinated with pulp science fiction magazines and by age eleven he began to write, imitating them in subject matter and style. By eighteen, he had sold his first story, and by twenty-one he had published “Robbie”—the first tale in the series gathered here—after several rejections. Over the next few years, he continued to test the Three Laws of Robotics in a series of robot stories. In 1941, inspired by Gibbon’s *Decline and Fall of the Roman Empire*, Asimov imagined writing about the rise and fall of future civilizations as if he were a historian looking back. He called such writing future-historical. The first story, “Foundation,” grew into a series that rivaled his robot stories for fame and influence. By 1949, the two series (and other writings), had cemented Asimov’s reputation as one of “The Big Three,” along with groundbreaking science fiction authors Arthur C. Clarke and Robert Heinlein. Even with early success, Asimov could not afford to be a full time writer until 1958. Meanwhile, he continued to write while obtaining his B.S. in 1939, M.A. in 1941, and Ph.D. in 1948 from Columbia University in the field of Chemistry. He taught at Boston University from 1949 to 1958 and remained a faculty member throughout his life. This “science” part of Asimov’s science fiction shows in his early commitment to make the science in his stories realistic (or at least plausible). In addition, throughout the 1960s and 1970s he primarily wrote non-fiction science works that covered a dazzling number of subjects including Astronomy, Earth Sciences, Physics, and Biology, among others. Calling Asimov a prolific writer would be an understatement. Through the years he tried his hands at literary criticism (from Shakespeare to Gilbert and Sullivan), humor (mainly limericks), children’s literature, autobiography, and editing. In addition, he managed to find time to write histories of Europe, North America, Greece, Egypt, England, and Earth. He wrote more than 1,600 essays and published at least 450 books. Famously, Asimov had at least one book published in each of the ten major Dewey Decimal library classifications. As he said in an interview, “I wrote everything I could think of.” Asimov won every major science fiction award during his life. He won seven Hugo awards, the first in 1963 and the last in 1995. He was also honored with

two Nebula awards. In 1986, the Science Fiction Writers Association named him a Grand Master and eleven years later he was inducted into the Science Fiction and Fantasy Hall of Fame. Asimov died in 1992 but his work lives on through new generations of readers, writers, and scientists. Rather than being outdated, his writing has proved prophetic. Reading his stories about robots in 1950, we would have thought that his reach exceeded his grasp. As advances in robotics and brain imaging have brought the idea of a human-like robot closer, we recognize that the day may come when we just might see Robbie tending to our own children.

DISCUSSION AND WRITING

1. Why are there no female robots in the stories?
2. The two groups consistently opposed to robots are labor unions and religious fundamentalists. Why did Asimov single out these groups to be threatened by robots? What do the groups share that makes them hostile towards robots? What other groups might not welcome robots into our world? Why? What groups would be happiest to see robots develop? Why?
3. What do Mike Donovan and Gregory Powell look like? Without letting them look at their books, have your students describe the two in as much detail as possible. What color are their eyes? How tall is each one? What race are they? Then ask your students to prove their descriptions from citations in the book. Asimov gives only one physical detail about the two (in “Catch That Rabbit”): Donovan has red hair.
4. In teams or individually, have your students check Asimov’s science. For example, when he claims that sound can’t travel on an airless asteroid (“Catch That Rabbit”), is he right? Why or why not? You can adjust this topic to your students’ level. For more exotic questions, they may try to figure out how fooling around with hyperspace might blow a hole in “the normal spacetime fabric” (“Little Lost Robot”).
5. Why does Robbie always want Gloria to tell him the story of Cinderella? If Robbie is a modern Cinderella, who are the wicked stepsisters, the fairy godmother, and the prince?
6. Early in “Liar!” RB-34 says that fiction helps him understand people better than science does. For him, “Science is just a mass of data plastered together by make-shift theory.” In what ways do your students agree with RB-34? How can a novel tell you more about humans than an anthropological textbook, or physics? What facts do novels leave out that science books include?
7. Asimov tended to be a pessimist about humanity. Late in life, he allowed the possibility that humans might improve in the future. “But still,” he said, “people tend to do things that harm Humanity.” Do your students agree or disagree with Asimov’s pessimism? Why or why not? What, for Asimov, makes robots “a better breed” than us? What, for your students, makes us a better breed than robots?
8. The stories often use scientific-sounding terms without explanation (e.g., “hyper-imaginary,” “Mitchell Translation Equation,” “positronic brain,” “hyperatomic travel,” “Planar reactions,” “etheric physics,” etc.). Have your students pick a few of them and invent stories—consistent with their context in *I, Robot*—that explain what these terms are and how they work.
9. Why do robots call humans “master” while humans generically refer to the robots as “boy”?

10. In “Evidence,” Asimov writes that “the three Laws of Robotics” are the essential guiding principles of a good many of the world’s ethical systems.” First, lead your class in a discussion of the concept of an ethical system. What is it? Can your students identify some? Then encourage your students to find out how many of the three Laws are represented in the systems they have identified, and in what order. Does the U.S. legal system, for example, require obedience over self-preservation? If so, why? If not, why not?

11. Through the 1940s, Asimov published each story in *I, Robot* as an individual story. In 1950, he collected the stories and published them together as a book. What clues can your students find in the book that show that the stories have been joined together? Where are the seams? What techniques did Asimov use to make the stories seem like one whole book? Where does this reweaving work well? Poorly? Why does it work in some places but not others?

12. This book begins with a story about a robot that is dominated by a little child (“Robbie!”) and ends with a story in which robots control every facet of human life (“The Evidable Conflict”). How likely do your students find a situation where humans would give up control of their worlds to machines? Would we give up the ability to own things? To determine our own movements? To what degree do they think we already have? What signs are there that our lives already have become controlled by machines? That we control our machines?

13. Asimov admits in his Memoir that, in his early writing, he was most comfortable with European-American characters. What signs of discomfort can your students detect when he writes non-European characters like Ching Hso-lin or Lincoln Ngoma (“The Evidable Conflict”)? Put another way, would Asimov have written any differently if Hso-lin (or others) had been Powell or Donovan? For example, would he have noted that Powell spoke in “precise English” as with Hso-lin, or that Donovan’s English was “colloquial and mouth-filling,” as with Ngoma?

14. Although most readers focus on the Three Laws of Robotics as the animating principle for the robot stories, there is another factor at work: emotional attachment. Asimov said, “Back in 1939, I realized robots were lovable.” What is lovable about the various robots in the stories? Which one was the most lovable? Why? Which was least lovable? Why? How does Asimov manage to make a hunk of metal lovable (or unlovable)?

15. How would the collection have changed if it were titled *Mind and Iron* (as Asimov wanted to call it originally)? What does the title, *I, Robot*, communicate that the title, *Mind and Iron*, doesn’t? Similarly, how would the first story change if it were titled “Strange Playfellow” instead of “Robbie?” What does strange playfellow setup that Robbie doesn’t? Come up with other titles that Asimov might have considered for the individual stories and the whole collection.

16. *I, Robot* has been turned into a major motion picture starring Will Smith. How does the movie compare with your book-reading experience? What do you think of the adjustments made and liberties taken when converting this collection of stories to one seamless film adaptation?

SUGGESTED ACTIVITIES

1. Have your students invent their own philosophical puzzle involving the Three Laws of Robotics using Asimov's human characters, but new robots. They might, for example, imagine a story where Powell and Donovan meet *Star Wars* character R2-D2 who is pulled in three directions by an order to destroy himself, the knowledge that destroying himself will kill a human, and the knowledge that not destroying himself will kill another human.

2. Asimov was deathly afraid of flying, but many of his stories involve flight across the earth to other planets and to distant galaxies. Have your students choose something they fear, and encourage them to write a science fiction story that involves that fear indirectly. For example, if someone is afraid of heights, s/he might write about a society that lives in the treetops. If someone is afraid of spiders, s/he might write about a society based on the pattern of a spider's web. After they write the story, have them consider how fear factored into their composition. Did they tend to write less about what they were afraid of? More? Did they write about their fear less directly? Return to Asimov's stories and see if you can identify the marks of fear when he writes about flight. (He was also afraid of other things that they might look for in a biography, or his memoir.)

3. Let your students pick a story by another author that involves robots and compare it to Asimov's. What similar concerns do they have? How human are the robots? What contrasts do they find between themes that interest Asimov and the other author?

4. What role do machines play in our lives today? Have your students keep a journal that lists every machine that helps them live their lives. A list might start, for example, with the alarm clock that wakes them up, the refrigerator that keeps the milk cold, the water heater that keeps the water hot, the computer that transmits email and stores their homework, the vehicle that drives them to school, the phones that deliver messages and pictures, and so on. What would life be like without these machines? In discussion, or writing, have them imagine a world where one by one, all these machines vanish. How would we eat, communicate, travel, etc.? Turning what they learn to the past, have your students research the history of a machine that has become indispensable to us today. What did people do before a particular machine was invented (e.g., clocks)? What changes happened when the machine was invented? Perfected? Turning toward the future, ask your students to think of machines that have yet to be invented. What things will become necessary to future generations that we do not have?

Robots for the Chemistry class

Topic: Robots

teacher: phd. SILVIA PETRESCU

Aims

- To practise dangerous chemical reactions
- To practise using robots in chemistry laboratory

Age group 13 – 19

Time 60 - 90 minutes

Materials

1. Robots student worksheet
2. Robotic hand
3. There are several websites that offer virtual chemistry laboratory simulations, such as:
 - ChemCollective (<https://chemcollective.org/simulations>): Provides a collection of virtual chemistry labs covering various topics, including acid-base reactions and redox reactions.
 - PhET Interactive Simulations (<https://phet.colorado.edu/en/simulations/category/chemistry>): Offers a wide range of interactive simulations for chemistry, including acid-base solutions and reactions.
 - Virtual Chemistry Experiments (<https://www.virtualchemistrylab.com>): Provides a platform for conducting virtual chemistry experiments, including acid-base titrations and reactions.

Introduction

Robots can be used in various ways in the field of chemistry to enhance research, experimentation, and analysis processes. Here are a few examples:

1. Automated synthesis: Robots can perform repetitive and precise tasks involved in chemical synthesis, such as measuring and dispensing reagents, mixing solutions, and controlling reaction parameters. This automation improves efficiency, accuracy, and reproducibility in chemical reactions.
2. High-throughput screening: Robots can be utilized to conduct large-scale screening of chemical compounds for drug discovery or material synthesis. They can handle multiple samples simultaneously, perform assays, and collect data, allowing for rapid and systematic screening of thousands or even millions of compounds.
3. Sample preparation: Robots can automate sample preparation techniques, such as sample dilution, extraction, and purification. This reduces human error and increases the speed and reliability of sample processing in analytical chemistry.
4. Analytical instrumentation: Robots can be integrated with analytical instruments like spectrometers, chromatographs, or mass spectrometers. They can load samples, run analyses, and record data, enabling high-throughput analysis and minimizing human intervention.

5. Laboratory logistics: Robots can assist in managing and organizing laboratory operations. They can transport samples, dispose of waste, and perform routine tasks like labeling, capping, or sorting chemicals. This helps streamline workflows and improve safety and efficiency in the laboratory.

It's important to note that the specific applications of robots in chemistry may vary depending on the available technology, **research requirements, and laboratory setups.**

Procedure

1. Introduce the topic - Experiment: Reaction of copper with nitric acid using a robotic arm.

- Attention! Demonstration experiment performed only by the teacher under the niche!
- Attention! Nitric acid is extremely caustic!
- Attention! The resulting gas, nitrogen dioxide is extremely toxic and corrosive!

Materials required:

Test tube, concentrated nitric acid, copper granules, pipette.

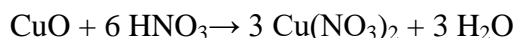
Robotic hand for experiment (students should not stand near the laboratory equipment during the experiment due to the danger of poisoning).

Description of the experiment:

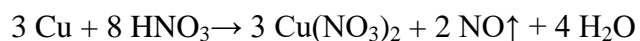
- Put some copper granules in the test tube.
- Pipet a few drops of concentrated nitric acid onto the copper.
- What do you notice? After a short time a red-brown gas is released.

Explanation

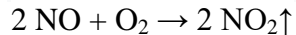
Copper reacts with nitric acid or with concentrated solutions of sulfuric acid, when heated, without substituting hydrogen. The reaction basically proceeds in two stages: the reaction of copper with the acid with the formation of oxide and water, followed by the transformation of the oxide into salts:



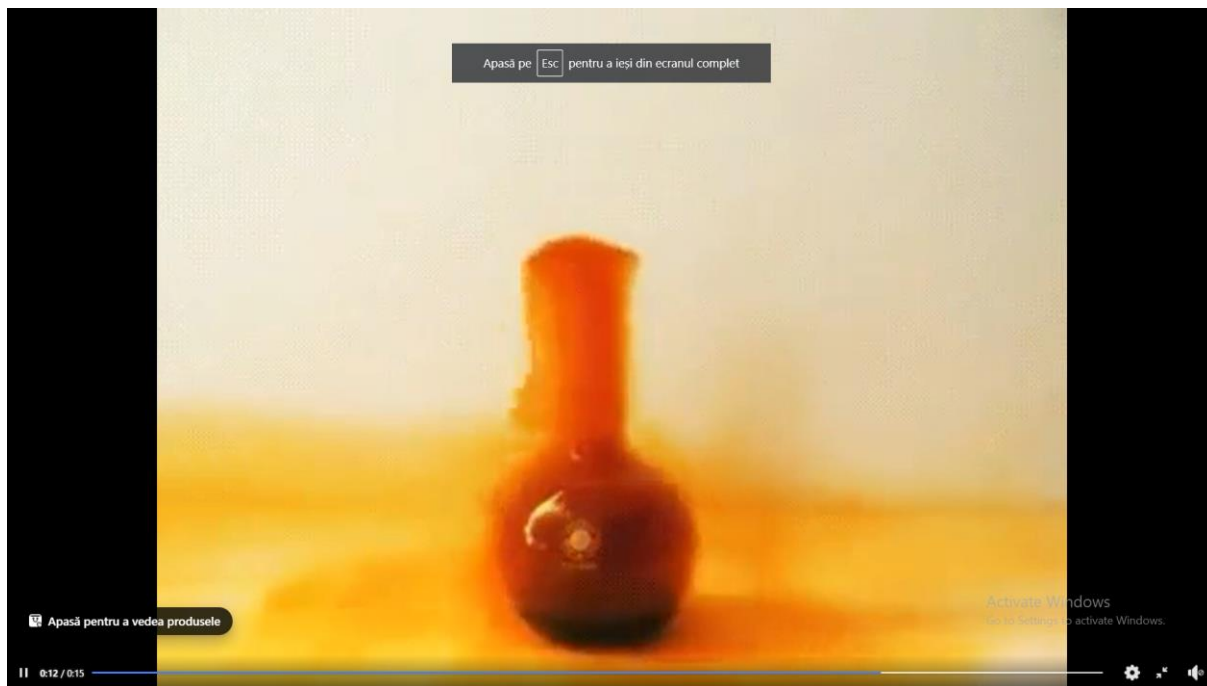
Global reaction:



The resulting nitrogen monoxide is oxidized to nitrogen dioxide:



2. Using websites for dangerous chemical reactions



<https://www.facebook.com/meditatiitop/videos/reactia-chimica-intre-cupru-si-acid-azotic-invata-cu-placere-la-centrul-de-medit/1026131697524392/>

<https://www.youtube.com/watch?v=NqJv7Q3Yfs8>

<https://www.youtube.com/watch?v=8hYLodjduJE>

3. Simulation and virtual modeling (CIROS[®], Robotino, SoftwareRoboCIM – LabVolt series)

The simulation and virtual modeling software ideally prepares for practical training in the learning laboratory. Thus, you will complete and expand the concepts of holistic learning in a meaningful and sustainable way.

Today's PC technology allows realistic 3D simulations, even for very complex systems.

Virtual simulation and modeling software supports today's changing work and learning habits. The focus is on visual understanding, the attractive presentation on the PC motivates and promotes the learning process.

The advantages of our simulation and virtual modeling software are:

- Virtual representation of physical training
- A good preparation for the learning workshop by working on virtual solutions
- Reducing the volume of work and time
- Reduction of the actually necessary teaching material
- The trainees are trained in handling and operating real hardware
- The students prepare and experiment virtually and independently of the location

Programming a robot

Aims

- To learn to program a robot using python
- To practise using robots in computers laboratory

Age group 13 – 19

Materials

1. Robots
2. Computers

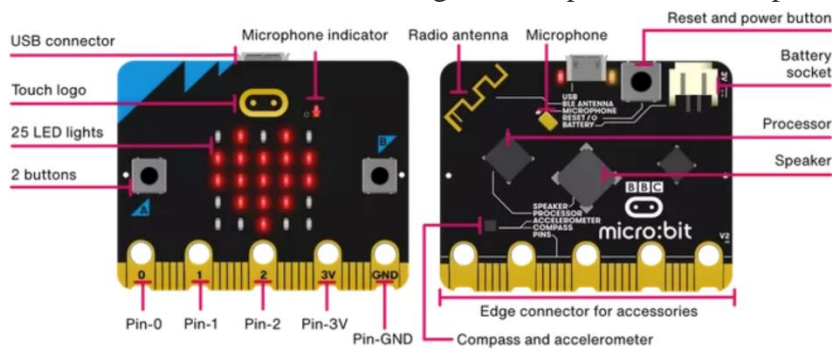
Teachers: Angelica and Cristian Toncea

Introduction in Microbit – 180 minutes

Microbit development board

The Microbit (BBC Micro Bit or micro:bit) is a microcontroller (a simple minicomputer with limited capabilities that can run a single program) developed with the help of the BBC for educational purposes. Key features of the board include: Nordic nRF52833 SoC

- ARM Cortex-M4 core processor with 64 MHz
- RAM memory: 128 KB
- Flash memory: 512 KB
- USB connection interface
- Microphone and speaker
- Radio antenna
- 2 programmable buttons
- 25 LEDs
- 3 programmable pins
- Sensors such as: touch, light, temperature, compass or accelerometer



Microbit simulator

To be able to simulate the interaction and programming of a Microbit board, we will use an online platform called MakeCode. The first steps we will take in getting comfortable with this simulator is to create a project and write a program using Visual Programming.

Visual Programming (Blockly)

Visual Programming or Blockly is a library developed by Google that allows us to write code using blocks that can be linked together using the mouse (drag-and-drop). Once the program is "assembled" and run, the equivalent code is generated in a high-level language such as Python or JavaScript (in the case of the MakeCode platform, the resulting code is in JavaScript).

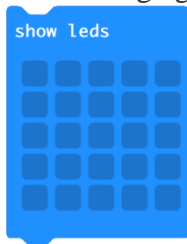
The advantage of using Blockly is that the level of programming knowledge required is low, and a minimal program can be built easily. The major disadvantage occurs when we want to build larger and more complex programs, its development in a normal programming language being much faster and easier.

Next we will see what are some of the main blocks within the MakeCode platform:

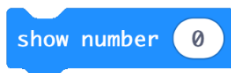
Basic blocks

These are simple blocks used to simulate a screen from the LEDs available on the board, but also other functions such as delays, which delay the execution of the next block of code. Some examples are:

Image generation using LEDs:



Display a number using the LEDs:



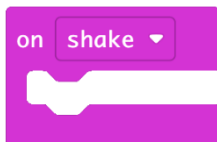
Delay the execution of the next code block:



Blocks for interacting with board functionalities

These are the blocks that interact with the sensors available on the board such as touch, gyroscope (for inclination) or temperature, but also with other components such as the two buttons on the board. Some examples are:

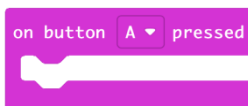
Detection of repeated tilting of the plate:



Touch detection (the sensor is under the logo):



Button press detection:



Repetitive loops

The section of blocks contained within this block will be repeated a specific number of times. The most important are:

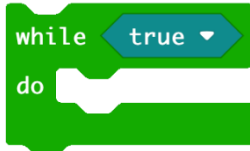
Repeating a section a repeated number of times:



Repeating a section a repeating number of times using a variable:



Conditional repetitive loop:



Interruption of a repetitive loop:



Logical blocks

Here we have blocks that execute the inner blocks only if a certain condition is met, but also blocks that return a logical value (true or false). The most important are:

Conditional section - executed if the value is true:



Checking the equality of two numbers:

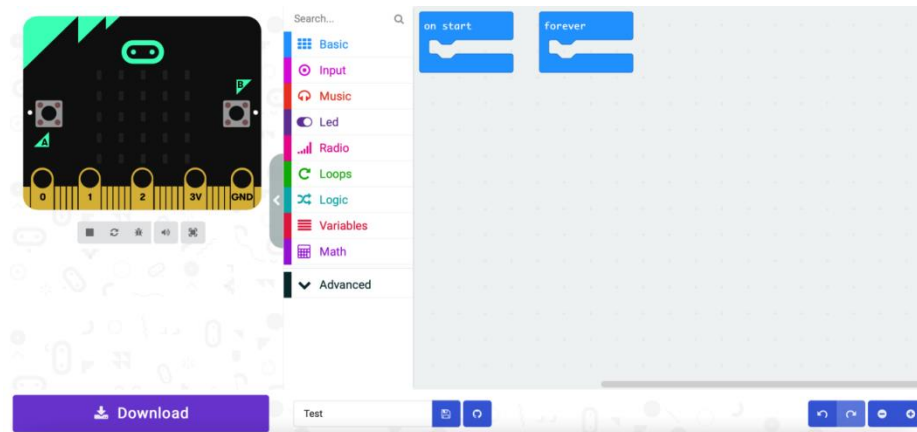


Checking the logical value of two values:



Interaction with the platform

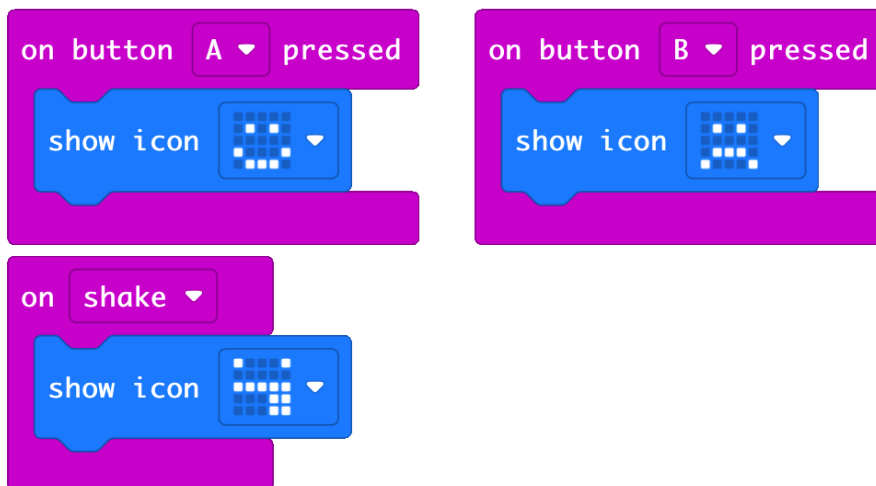
To create a new project, go to the MakeCode platform page and press the "New Project" button. We will then write a name for the new project and then press "Create". Below we see what the platform looks like when we create a new project:



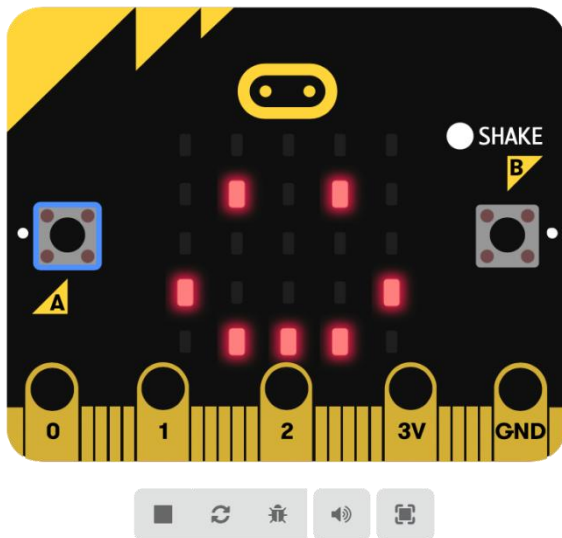
On the left side we have the plate with which we can interact (either by pressing one of the two buttons or the logo, or by tilting the plate). In the central part we have the blocks that we can use grouped in folders, which we can use with "drag-and-drop" and then we can join them. On the right side we have the place where we compose the program from blocks.

The platform does not store the projects we write, it uses the storage facilities of the browser. Therefore, to be able to use the projects later, we must download them using the "Download" button. The project will save locally as a .hex file. Later, on the main page, instead of creating a new project, we have the option to select "Import" and upload the .hex file.

An example of a program that can be written is the following:



This code will generate using the LEDs on the board a certain type of face depending on the interaction we have with the board on the left side of the browser. For example, if we press the A button, we will get a smiley face like below:



Exercises

After solving an exercise, save the local project with a suggestive name so you don't lose what you've been able to work on and can access the solutions later. For each exercise create a new project.

1. Create a new project where you reproduce the code given above in the example. You can delete the "on start" and "on forever" blocks created automatically when the project is generated.
2. Simulate rolling a dice: when you "shake" the board, a random number between 0 and 6 will be displayed. Hint: The "Math" block group
3. Generating a sequence using the LED screen. You will first generate a picture, and after a 2 second delay, you will write your first name (preferably abbreviated) or a nickname. Finally after another 200 milliseconds delay, you will light up all the leds on the board (simulating filling the screen).
4. Depending on the button pressed on the board generate an image to be displayed.
5. Sound Play: If a sound is generated, the LED screen will light up for two seconds. The moment it is quiet again, the LED screen will be blank again. Hints: "Input" block group and variables.
6. Generate a "level" bar from the LED display, reflecting the brightness level. Hint: the "Input" group.
7. Generate a sequence of images on the screen at a preset interval.
8. Create a timer. Pressing the A button will start the timer and display an image on the screen, and pressing the B button will stop the timer and the number of seconds will be displayed on the screen. Hint: block group "Input" to "More" and group "Math".

9. Depending on the interaction with the board (pressing a button, the logo or "shaking" the board) play a different sound and display an image on the screen.

10. Smart light: If the light level drops below a certain threshold, the screen will be fully illuminated; if the level rises above a certain threshold the screen will fully light up. Moreover, a corresponding sound for day/night will also be played.

11. Advanced Exercise: Pet

a. The animal is represented by a smiling or angry face displayed on the screen.

b. To "pet the animal" press the logo. A corresponding sound will be played when the logo is pressed.

c. A timer will be started as soon as the program starts, and if 10 seconds have passed since it was not stroked, the "animal" will make a corresponding sound. If 20 seconds pass since it has not been petted, the "pet dies" and an "X" or other representative image is displayed on the screen (be creative).

d. The timer is reset each time the logo is pressed. After solving an exercise, save the local project with a suggestive name so you don't lose what you've been able to work on and can access the solutions later. For each exercise create a new project.

Introduction in Python – 120 minutes

Purpose

- familiarization with basic programming concepts
- interaction with the development board using Python

Keywords

- for, if, while, vector
- basic, music
- show_string, show_number, show_leds, clear_screen, pause
- play_tone, start_melody

Programming concepts

Variables

Variables are ways we reference values of certain types. Unlike programming languages like C or C++, in python data types are identified at initialization time.

Some examples of types in python are:variable.py

```
x = 2                #int - integer
x = 2.7              #float - number with decimals
x = "c"              #char - ASCII character
x = True             #bool -logical value: True/False
```

Once a data type has been assigned, we can reassign another type through a new initialization (as above), but some data types cannot be combined, for example: adding an integer to a string.

Strings / Vectors

Strings/Vectors are smarter variables of a certain size that can store a string of values of each type above. Example:

```
variable.py
x = "Hello Python!!!"          #string - şir de caractere,
    declarat direct folosind ""
x = ['h', 'e', 'l', 'l', 'o']  #string - declarat ca vector de
    caractere ASCII
x = [2, 3.2, 21, 12]           #array - vector de numere
```

One atypical thing about Python compared to many classical programming languages is that it allows us to have values from different data types within the same vector (integers, strings and booleans).

If - Else

The if statement is a conditional statement used to execute statements based on certain conditions. The if-else construction translates as "if - otherwise":

if_else.py

```
a = 10
if a == 10:           # Dacă a este egal cu 10
    a = a / 2        # Fă operații dacă a este 10
else:                 # Altfel
    a = a * 2        # Fă operații când a este diferit de 10
```

The block of code to be executed is identified by the number of lines, so be very careful when indenting your code!

For

The for structure is a repeating structure depending on a condition. Way of usage:

for.py

```
fruits = ["măr", "cireșe", "pară", "banană"]

# for element in vector_of_elements:
for f in fruits:
    print(x)

# The code above will display: apple cherry pear banana

# Alternatives closer to C:

# for element in range(maximum value that will not be reached):
for x in range(6):
    print(x)

# The code above will display: 0 1 2 3 4 5

# for element in range(initial value, maximum value that will not be reached,
increment value)
# The last value may be missing from the build, the default value being 1
for x in range(2, 10, 2):
    print(x)

# The code above will display: 2 4 6 8
```

Looping through a string

Accessing a certain value from a vector is done through the index (position) of the value in the vector.

```
vector.py
```

```
v = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
sum = 0

print(v[3])           # display the value at position 4: 4 (indexing is done
starting from 0)
for i in range(len(v)):
    sum = sum + v[i]

print(sum)           # display 55

# Another alternative for calculating the amount would have been:
for element in v:
    sum = sum + element
```

Interacting with the simulator in Python

To be able to interact with the simulator, each of the functions exposed in Python by the board must be prefixed with "basic" (which is the object in which the respective functions are located).

An example of such a function is writing a string to the screen:

```
basic.show_string('Hello!')
```

Interaction with the leds on the board

The main functions we have available are:

Display a string on the screen:

```
basic.show_string('Hello!')
```

Displaying a number on the LED screen:

```
basic.show_number(0)
```

Displaying an image from LEDs:

```
basic.show_leds("""
.#.#.
#####
#####
#####""")
```

```
.###.  
..#..  
""")
```

```
# Se va afișa o inimă din Led-uri  
## - semnifică led aprins  
# . - semnifică led stins
```

Switching off all LEDs on the board:

```
basic.clear_screen()
```

Delay program execution by a certain number of milliseconds:

```
basic.pause(1000)
```

Sounds using Python

Features for playing sounds using the Microbit simulator:

Playing a single note using the "Note" object

```
freq = Note.D  
music.play_tone(Note.C, 1000)
```

Play a song only once:

```
melody = music.built_in_melody(Melodies.ENTERTAINER) # Vom folosi o melodie  
predefinită  
music.start_melody(melody, MelodyOptions.ONCE) # Redă o singură dată melodia
```

Playing a sample until that sample ends:

```
soundExpression.giggle.play_until_done()
```

Exercises

1. Generate a sequence of images to appear on the screen at a certain time interval.
2. Generate a sequence using the LED display. You will first generate a picture, and after a 2 second delay, you will write your first name (preferably abbreviated) or a nickname. Finally after another 200 milliseconds delay, you will light up all the leds on the board (simulating filling the screen).
3. Display the numbers from 1 to 10 on the LED screen.
4. Display the even numbers from 1 to 20 on the screen.
5. Cycle through the numbers from 1 to 30: if the number is divisible by 5, show a heart, if the number is divisible by 3 (but not 5), show a smiley face, otherwise show a sad face.

6. Generate a vector of 20 random elements between 0 and 100. Then loop through the generated vector and count the number of elements that are divisible by 10 and the number of elements that are divisible by 9. If the number of elements that are divisible by 10 is greater, display a smiley face on the screen. Otherwise, show a sad face. Hint: for generating random elements, look in the Math section at the `randint` function.
7. Modify the previous project to find the largest prime number in the vector and it will be displayed on the screen. If there is no prime number in the vector, a sequence will be generated where all the LEDs light up 3 times with a 500ms delay between them.
8. Modify the previous project so that in case of success the song "Entertainer" will be played, and if the number is not found the song "Funeral" will be played. The `music.built_in_melody` function will be used.
9. Google the notes for the song "Twinkle Twinkle little star" and create a vector with those notes. Then use the `play_tone` function to play the song.
10. Adapt the previous project to generate a different image for each note that is played.
11. Create a new project in which you will play the same melody, using the function `music.play_melody` (more details in the platform, in the music menu, first block). Try adapting the code to generate images similarly to the previous exercise.
12. Implement playback of the first notes of your favorite song. Change various variables like tempo and volume.
13. Play a sample like the one in the lab and show a smiley face.

Purpose

- interaction with the existing buttons on the board
- using the sensors present on the board

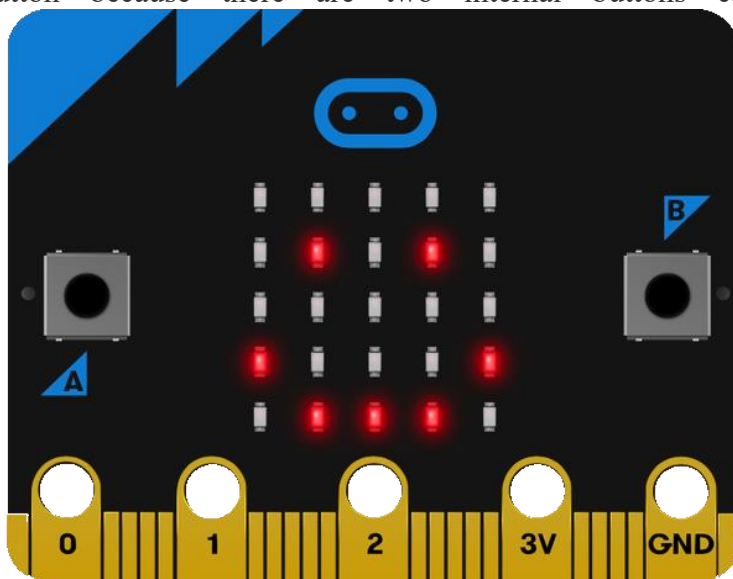
Keywords

- button, sensor
- `button_is_pressed`, `on_button_pressed`
- acceleration, `light_level`, `compass_heading`, temperature

Buttons

A button is conceptually speaking like a switch that allows or not allows current to flow through a certain section of a circuit. So to see if a button (or a switch) is pressed or not, we check the voltage value on that section of the circuit. If the logic value is 0, then the button is not pressed (basically, the circuit is broken), and if the logic value is 1, then the button is pressed. The actual voltage value can be 5V or 3.3V depending on the development board used and the type of pin used to connect that button.

For the Microbit board and within the simulator we will not need to connect any external button because there are two internal buttons called “A” and “B” available:



For button interaction we have the following main functions available:

- Checking the status of a button whether it has been pressed or not:

```
value = input.button_is_pressed(Button.A) # return True or False depending on
# button A status.
# for button B, we will use Button.B
```

- Execution of a section of code when a certain button is pressed:

```
# first, we define a function that contains the section of code that will be executed
# when the button is pressed
def on_button_pressed_a():
```

```
# here we write the code
```

```
pass
```

```
# we set the function that will be called when the A button is pressed
```

```
# for button B, we'll use Button.B
```

```
input.on_button_pressed(Button.A, on_button_pressed_a)
```

Sensors

A sensor is a device that measures a physical parameter of the external environment (such as light, humidity, temperature or sound) and converts it into a signal that can be interpreted by the device. Often the sensor is a variable resistor that changes in direct proportion to the physical parameter being measured. Thus, similarly to a button, to read the value of a sensor we will read the value of the voltage on the sensor.

The main difference to the button is that we need an analog pin. Unlike digital pins that can only be used for 0V or 5V values, analog pins can be used to read or write a larger range of values and are useful for fine measurement. Write values can be between 0-255 and are mapped to 0 - 5V or 0 - 3.3V depending on the development board and pin.

On the Microbit board and in the simulator available online, we have at our disposal several sensors such as: light, temperature, gyroscope, compass or noise.

To interact with the sensors available on the board, we have the following main functions at our disposal:

- Acceleration value on an axis:

```
accel_x = input.acceleration(Dimension.X) # values between -1023, 1023
```

- Light intensity value:

```
light_val = input.light_level() # values between 0 (dark) and 255 (light)
```

- Orientation on the compass:

```
light_val = input.compass_heading() # values between 0 and 359 degrees  
# 0 - 44: Nord  
# 45 - 134: Est  
# 135 - 224: Sud  
# 225 - 314: West  
# 315 - 359: Nord
```

- emperature value in degrees Celsius:

```
temp = input.temperature()
```

- Touch sensor:

```
touched = input.logo_is_pressed() # returns True is logo is pressed  
# and False otherwise
```

- Noise sensor:

```
noise_level = input.sound_level()      # value between 0 and 255
```

- Running a code sequence when "shaking" the board:

```
# first, we define a function that contains the section of code that will be executed  
# when the board will "shake".  
def on_gesture_shake():  
    # here we write the code  
    pass  
  
# we set the function that will be called when  
# the board will detect that it is shaken.  
input.on_gesture(Gesture.Shake, on_gesture_shake)
```

- Running a code sequence when a noise is detected:

```
# the function definition that contains the section of code to be executed  
def on_sound_loud():  
    # here we write the code  
    pass  
  
# we set the function that will be called when  
# will be detected:  
input.on_sound(DetectedSound.LOUD, on_sound_loud)
```

- Running a code sequence when the logo is pressed:

```
# the function definition that contains the section of code to be executed  
def on_logo_event_pressed():  
    # here we write the code  
    pass  
  
# we set the function that will be called when  
# will be detected:  
input.on_logo_event(TouchButtonEvent.PRESSED, on_logo_event_pressed)
```

Exercises

All exercises will be done on MakeCode

1. Simulate rolling a dice: when you "shake" the board, a random number between 0 and 6 will be displayed.
2. Music buttons: depending on the button pressed, play a song generated from the notes (as in the previous lab)

3. Depending on which button is pressed on the board, generate an image to be displayed on the LED screen.

4. Modify the previous application so that when a noise occurs, all the LEDs light up for 2 seconds and then turn off.

5. Smart light: if the light level drops below a certain threshold, the screen will be fully illuminated; if the level rises above a certain threshold the screen will fully light up.

6. Interactive thermometer: the temperature value will be read, and depending on the range in which the value is read, an action will be taken:

- 0 - 30 degrees Celsius - a skull will be displayed and the built-in song "Funeral" will be played.
- 30 - 37 degrees Celsius - a smiley face will be displayed and the built-in song "Entertainer" will be played.
- 37 - 42 degrees Celsius - the "-_-" face will be displayed (ASLEEP in the predefined ICONs) and the built-in song "POWER_DOWN" will be played
- 42 - 50 degrees Celsius - a skull will be displayed and the built-in song "Funeral" will be played.
- Optionally, the "Funeral" song can be replaced with the generation of the "Astronomia" song from the notes.

7. Implement a counter to display on the screen how many times the logo has been pressed.

8. Sound Compass: In a while True, depending on the 4 cardinal points and the orientation of the board, play a different song at a time and display the cardinal point on the screen.

9. Clap counter: Count how many times a loud clap is generated and display the counter on the screen. The counter will reset when it reaches 10.

10. Advanced Exercise: Pet

a. The animal is represented by a smiling or angry face displayed on the screen.

b. To "pet the animal" press the logo. A corresponding sound will be played when the logo is pressed.

c. A timer will be started as soon as the program starts, and if 10 seconds have passed since it was not stroked, the "animal" will make a corresponding sound. If 20 seconds pass since it has not been petted, the "pet dies" and an "X" or other representative image is displayed on the screen (be creative).

d. The timer is reset each time the logo is pressed.

Applications

Purpose

□ deepening the knowledge acquired in the first 3 lessons and laboratories

I. Smart home

In this project we will simulate the functionality of a smart home. The main facilities we want are:

- Intelligent lighting: depending on the brightness level, lines of LEDs will gradually light up (High light → all LEDs off, Dark → all LEDs on)
- Ambient Music: Set 3 brightness level ranges within which a song will be played only once. The songs must be created by you from the notes (you can choose any song to generate).
- Sound-controlled lighting: for each clap (or loud noise), a line of LEDs will light up.
- Display of ambient parameters: upon repeated presses on the logo, the values of the ambient parameters (temperature, brightness, noise level), but also the other values from the sensors: acceleration and compass, will be displayed on the LED screen.
- In case of an earthquake (when the slab shakes), a police siren will sound.
- Temperature variation graph: display in the console using print the temperature value once per second.
- Random display: At the beginning of the program, manually create a vector of at least 10 favorite Icons (like HEART). In the main program run loop, regularly check how much time has passed since the board was booted (running_time from Input). At every multiple of 5 seconds since the execution of the code started on the board, display a random Icon from the created vector.

II. Drawable interactive screen

In this project we will create a board on which we can draw using dots. When we press the logo (touch sensor), a lighted led will be generated in the lower left corner. Using the A button, we can "move" the lit led up on the column it is currently on. Using the B button, we can "move" the led to the right on the line it is on now. Once we have reached the position where we want to leave the led lit, we press the logo once more. To generate a new "lit led", press the logo once more.

Additional Requirements:

- Columns and lines are considered circular: once we have reached the end of the column/line, the led will return to the beginning of the column/line
- We cannot have two LEDs on in the same position; when I move a lit led over an already lit led, it will "jump" to the first free position in the direction of movement

Advanced Requirements:

- If when the point is generated, there is already a point in the lower left corner of the screen, it will be generated on the first free position on the first line. If the line is fully occupied, it will move to the next line.
- Extinguishing an LED: If there is already a lit LED on the position where we "moved" an LED, it will turn on and off repeatedly (once every 100 ms). The user will have two actions available:

1. Either press the logo and the led will turn off.

2. Either press one of the two buttons (A or B), and the led will move to the new position. Thus the "previously superimposed" led will remain on.

GLOSSARY OF TERMS - ROBOTICS

Absolute Data (ABSO Data)

Absolute Data (ABSO Data) is a correction factor for data that establishes an indicated value of zero when the robot is at the predetermined Home (calibration position).

Accuracy

Accuracy is the measurement of the deviation between the command characteristic and the attained characteristic (R15.05-2), or the precision with which a computed or calculated robot position can be attained. Accuracy is normally worse than the arm's repeatability. Accuracy is not constant over the workspace, due to the effect of link kinematics.

Active Compliant Robot

An active compliant robot is one in which motion modification during the performance of a task is initiated by the control system. The induced motion modification is slight, but sufficient to facilitate the completion of a desired task.

Actual Position

The position or location of the tool control point. Note that this will not be exactly the same as the demand position, due to a multitude of unsensed errors, such as link deflection, transmission irregularity, tolerances in link lengths, etc.

Actuator

A power mechanism used to effect motion, or maintain position of the robot (for example, a motor which converts electrical energy to effect motion of the robot) (R15.07). The actuator responds to a signal received from the control system.

Arm

An interconnected set of links and powered joints comprising a robot manipulator that supports and/or moves a

wrist and hand or end-effector through space. The arm itself does not include the end-effector.

See Manipulator, End-effector and Wrist.

Articulated Manipulator

A manipulator with an arm that is broken into sections (links) by one or more joints. Each of the joints represents a degree of freedom in the manipulator system and allows translation and rotary motion.

Articulation

Describes a jointed device, such as a jointed manipulator. The joints provide rotation about a vertical axis, and elevation out of the horizontal plane. This allows a robot to be capable of reaching into confined spaces.

Assembly Robot

A robot designed specifically for mating, fitting, or otherwise assembling various parts or components into completed products. Primarily used for grasping parts and mating or fitting them together, such as in assembly line production.

Automatic Measurement Function

For optimal robot motion, the mass properties of the end-effector should be specified. These properties can be derived from a CAD model of the tool. The Automatic Measurement Function is an alternative to a CAD model, and it uses the robot arm itself to measure the tool properties. With this function, the user can register the load of tool, the position of the tool's center of gravity and the moment of inertia at the center of gravity.

Automatic Mode

See Play Mode.

Axis

A direction used to specify the robot motion in a linear or rotary mode.

Axis Interference

The Axis Interference Area is a function that judges the current position of each axis and outputs a signal based on whether the current position is within a predefined range.

Base

The stable platform to which an industrial robotic arm is attached.

Base Coordinate System

The Base Coordinate System (sometimes referred to as World Coordinate System) defines a common reference point for a cell or application. This is useful when using multiple robots or devices as positions defined in Base Coordinates will be the same for all robots and devices. (see figure at right)

Base Link

The stationary base structure of a robot arm that supports the first joint.

Burn-in

Burn-In is a robot testing procedure where all components of the robot are operated continuously for an extended period of time. This is done to test movement and movement programming of the robot at early stages to avoid malfunctions after deployment.

Computer Aided Design (CAD)

Computer Aided Design (CAD). Computer graphic applications designed to allow engineering of objects (or parts), which are to be manufactured. A computer is used as a tool to design schematics and produce blueprints, which enable the accurate production of the object. The CAD system enables the three dimensional drawings of basic figures, exact sizing and placement of components, making lines of specified length, width, or angle, as well as satisfying varying geometric shapes. This system also allows the designer to test a simulated part under different stresses, loads, etc.

Carousel

A rotating platform that delivers objects to

a robot and serves as an object queuing system. This carousel delivers the objects, or work pieces, to the loading/unloading station of the robot.

Cartesian Coordinates

Cartesian Coordinates is a type of coordinate system that specifies the location of a point in two dimensional space by a pair of numerical numbers, which further specify the distance to fixed axes that are perpendicular to each other. In simple terms, an XY graph represents a two dimensional Cartesian Coordinate System. When a point is specified in a three dimensional space (XYZ graph), it constitutes a three dimensional Cartesian coordinate system. A robot's TCP position is specified in a Cartesian Coordinate.

Cartesian Manipulator

A Cartesian Manipulator is a robot arm with prismatic joints, which allows movement along one or more of the three-axes in the X, Y, Z coordinate system.

Cartesian Topology

A topology, which uses prismatic joints throughout, normally arranged to be perpendicular to each other.

Cartesian-coordinate Robot

A Cartesian-coordinate Robot is a robot whose manipulator-arm degrees of freedom are defined by Cartesian Coordinates. This describes motions that are east-west, north-south and up-down, as well as rotary motions to change orientation.

Category 3 (Cat3)

Category 3 (Cat 3) means that the safety related parts of the control system will be designed so that:

- Single faults will not prevent the safety function from working correctly.
- Single faults will be detected at or before the next demand of the safety function.
- When a single fault does occur, a safe state shall be maintained until the detected fault is corrected.

- All reasonably foreseeable faults are detected.

Centrifugal Force

When a body rotates about an axis other than one at its center of mass, it exerts an outward radial force called centrifugal force upon the axis, which restrains it from moving in a straight tangential line. To offset this force, the robot must exert an opposing torque at the joint of rotation.

Circular Motion Type

A calculated path that the robot executes, and is circular in shape.

Clamp

An end-effector which serves as a pneumatic hand that controls the grasping and releasing of an object. Tactile, and feed-back force sensors are used to manage the applied force to the object by the clamp. See End-effector.

Clamping

The maximum permissible force acting on a body region, resulting from a robot collision where the period of contact results in a plastic deformation of a person's soft tissue.

Clamping Force

When contact can cause a body part(s) to be clamped.

Closed-loop

Control achieved by a robot manipulator by means of feedback information. As a manipulator is in action, its sensors continually communicate information to the robot's controller, which is used to further guide the manipulator within the given task. Many sensors are used to feed back information about the manipulator's placement, speed, torque, applied forces, as well as the placement of a targeted moving object, etc. See Feedback.

Collaborative Robot

Term used to describe a robot system designed to operate in one or more of the four collaborative modes.

- Safety Monitored Stop

- Hand Guiding
- Speed and Separation Monitoring
- Power and Force Limiting

Command Interpreter

A module or set of modules that determines what the received command means. The command is broken down into parts (parsed) and processed.

Command Position

The endpoint position of a robot motion that the controller is trying to achieve.

Compliance

Displacement of a manipulator in response to a force or torque. A high compliance means the manipulator moves a good bit when it is stressed. This is called spongy or springy. Low compliance would be a stiff system when stressed.

Compliant Robot

A robot that performs tasks, with respect to external forces, by modifying its motions in a manner that minimizes those forces. The indicated or allowed motion is accomplished through lateral (horizontal), axial (vertical) or rotational compliance.

Configuration

The arrangement of links created by a particular set of joint positions on the robot. Note that there may be several configurations resulting in the same endpoint position.

Contact Sensor

A device that detects the presence of an object or measures the amount of applied force or torque applied on the object through physical contact with it. Contact sensing can be used to determine location, identity, and orientation of work pieces.

Continuous Path

Describes the process where by a robot is controlled over the entire path traversed, as opposed to a point-to-point method of traversal. This is used when the trajectory of the end-effector is most important to provide a smooth movement, such as in spray painting etc. See Point-to-Point.

Control Algorithm

A monitor used to detect trajectory deviations in which sensors detect such deviations and torque applications are computed for the actuators.

Control Command

An instruction fed to the robot by means of the human-to-machine input device. See Pendant (Teaching). This command is received by the robot's controller system and is interpreted. Then, the proper instruction is fed to the robot's actuators, which enable it to react to the initial command. Many times, the command must be interpreted with the use of logic units and specific algorithms. See Input Device and Instruction Cycle.

Control Device

Any piece of control hardware providing a means for human intervention in the control of a robot or robot system, such as an emergency-stop button, a start button, or a selector switch. (R15.06)

Control Mode

The means by which instructions are communicated to the robot.

Controllability

The property of a system by which an input signal can take the system from an initial state to a desired state along a predictable path within a predetermined period of time.

Controller

An information processing device whose inputs are both the desired and measured position, velocity or other pertinent variables in a process and whose outputs are drive signals to a controlling motor or actuator. (R15.02)

Controller System

The robot control mechanism is usually a computer of some type, which is used to store data (both robot and work environment), and store and execute programs, which operate the robot. The Controller System contains the programs, data, algorithms; logic analysis, and

various other processing activities, which enable it to perform. See Robot.

Coordinate System or Frame

A Coordinate System (or Frame) defines a reference position and orientation from which a robot position can be measured. All robot positions are defined with reference to a Coordinate System. Yaskawa robots utilize the following

Coordinate Systems:

- Base coordinates
- Robot coordinates
- User coordinates
- Cartesian coordinates

Central Processing Unit (CPU)

The Central Processing Unit (CPU) is the main circuit board and processor of the Controller System.

Cubic Interference Area

This area is a rectangular parallelepiped, which is parallel to the base coordinate, robot coordinate or user coordinate. The YRC1000 controller judges whether the current position of the manipulator's TCP is inside or outside this area, and outputs this status as a signal.

Cycle

A single execution of a complete set of moves and functions contained within a robot program.

Cyclic Coordinate System

A coordinate system that defines the position of any point in terms of an angular dimension, a radial dimension and a height from a reference plane. These three dimensions specify a point on a cylinder.

Cyclo Drive

A brand name for a speed reduction device that converts high speed low torque to low speed high torque, usually used on the major (larger) axis.

Cylindrical Topology

A topology where the arm follows a radius of a horizontal circle, with a prismatic joint

to raise or lower the circle. Not popular in industry.

Dead Man Switch

Deprecated term. See Enabling Device.

Degrees of Freedom

The number of independent directions or joints of the robot (R15.07), which would allow the robot to move its end effector through the required sequence of motions. For arbitrary positioning, 6 degrees of freedom are needed: 3 for position (left-right, forward-backward and up- down), and 3 for orientation (yaw, pitch and roll).

Direct-drive

Joint actuation, including no transmission elements (i.e., the link is bolted onto the output of the motor.)

Downtime

A period of time, in which, a robot or production line is shut down, due to malfunction or failure. See Uptime.

Drive

A speed (gear) reducer to convert high speed low torque to low speed high torque. See Harmonic Drive, Cyclo Drive and Rotary Vector Drive).

Drop Delivery

A method of introducing an object to the workplace by gravity. Usually, a chute or container is so placed that, when work on the part is finished, it will fall or drop into a chute or onto a conveyor with little or no transport by the robot.

Dynamics

The study of motion, the forces that cause the motion and the forces due to motion. The dynamics of a robot arm are very complicated as they result from the kinematical behavior of all masses within the arm's structure. The robot arm kinematics are complicated in themselves.

Emergency Stop

The operation of a circuit using hardware-based components that overrides all other robot controls, removes drive power from

the robot actuators, and causes all moving parts to stop.

Enable Switch

See Enabling Device.

Enabling Device

A manually operated device which when continuously activated, permits motion. Releasing the device shall stop robot motion and motion of associated equipment that may present a hazard.

Encoder

A feedback device in the robot manipulator arm that provides current position (and orientation of the arm) data to the controller. A beam of light passes through a rotating code disk that contains a precise pattern of opaque and transparent segments on its surface. Light that is transmitted through the disk strikes photo-detectors, which convert the light pattern to electrical signals. See Feedback, Closed-loop Control and Feedback Sensor.

EOAT

See Gripper or End-effector.

End-effector

An accessory device or tool, specifically designed for attachment to the robot wrist or tool mounting plate to enable the robot to perform its intended task. (Examples may include: gripper, spot weld gun, arc weld gun, spray point gun or any other application tools.)

Endpoint

The nominal commanded position that a manipulator will attempt to achieve at the end of a path of motion. The end of the distal link.

Error

The difference between the actual response of a robot and a command issued.

Expandability

Being able to add resources to the system, such as memory, larger hard drive, new I/O card, etc.

External Force Limit

The threshold limit where the robot moves to or retains position, even when external forces are applied (provided that forces do not exceed limits that would cause an error).

Feedback

The return of information from a manipulator or sensor to the processor of the robot to provide self-correcting control of the manipulator.

Feedback Control

A type of system control obtained when information from a manipulator or sensor is returned to the robot controller in order to obtain a desired robot effect.

Feedback Sensor

A mechanism through which information from sensing devices is fed back to the robot's control unit. The information is utilized in the subsequent direction of the robot's motion.

Flexibility

The ability of a robot to perform a variety of different tasks.

Force Feedback

A sensing technique using electrical signals to control a robot end-effector during the task of the end-effector. Information is fed from the force sensors of the end-effector to the robot control unit during the particular task to enable enhanced operation of the end-effector.

Force Sensor

A sensor capable of measuring the forces and torque exerted by a robot and its wrist. Such sensors usually contain strain gauges. The sensor provides information needed for force feedback.

Forward Kinematic Solution

The calculation required to find the endpoint position, given the joint positions. For most robot topologies this is easier than finding the inverse kinematic solution.

Forward Kinematics

Computational procedures which determine where the end-effector of a robot is located in space. The procedures use mathematical algorithms along with joint sensors to determine its location.

Frame

A coordinate system used to determine a position and orientation of an object in space, as well as the robot's position within its model.

Functional Safety Unit (FSU)

The Functional Safety Unit (FSU) is a component of the Yaskawa robot controller that provides programmable safety functions that enable collaborative operation of the robot. As these safety functions are programmable, the FSU allows the minimization of nearby overall equipment footprint, as well as human accessible areas. The FSU consists of two parallel Central Processing Units (CPUs) run concurrently, thereby providing dual channel checking. In addition, the FSU acquires robot position from its encoders independently from the motion control system of the robot. Based on this feedback, the FSU monitors the manipulator and tool's position, speed and posture.

Gantry

An adjustable hoisting machine that slides along a fixed platform or track, either raised or at ground level along the X, Y, Z axes.

Gantry Robot

A robot which has three degrees of freedom along the X, Y and Z coordinate system. Usually consists of a spooling system (used as a crane), which when reeled or unreel provides the up and down motion along the Z axis. The spool can slide from left to right along a shaft which provides movement along the Z axis. The spool and shaft can move forward and back along tracks which provide movement along the Y axis. Usually used to position its end effector over a desired object and pick it up.

Gravity Loading

The force exerted downward, due to the weight of the robot arm and/or the load at the end of the arm. The force creates an error with respect to position accuracy of the end effector. A compensating force can be computed and applied bringing the arm back to the desired position.

Gripper

An end effector that is designed for seizing and holding (ISO 8373) and "grips" or grabs an object. It is attached to the last link of the arm. It may hold an object using several different methods, such as: applying pressure between its "fingers", or may use magnetization or vacuum to hold the object, etc. See End-effector.

Hand

A clamp or gripper used as an end-effector to grasp objects. See End-effector, Gripper.

Hand Guiding

Collaborative feature that allows an operator to hand guide the robot to a desired position. This task can be achieved by utilizing additional external hardware mounted directly to the robot or by a robot specifically designed to support this feature. Both solutions will require elements of functional safety to be utilized. A risk assessment shall be used to determine if any additional safeguarding is necessary to mitigate risks within the robot system.

Harmonic Drive

Compact lightweight speed reducer that converts high speed low torque to low speed high torque. Usually found on the minor (smaller) axis.

Harness

Usually several wires, bundled together to deliver power and/or signal communications to/from devices. For example, the robot motors are connected to the controller through a wire harness.

Hazardous Motion

Unintended/unexpected robot motion that may cause injury.

Hold

A stopping of all movements of a robot during its sequence, in which some power is maintained on the robot. For example, program execution stops, however power to the servo motors remain on, if restarting is desired.

Home Position

A known and fixed location on the basic coordinate axis of the manipulator where it comes to rest, or to an indicated zero position for each axis. This position is unique for each model of manipulator. On Motoman® robots there are indicator marks that show the Home position for the respective axis.

IEC

International Electrotechnical Commission

Inductive Sensor

The class of proximity sensors, which has half of a ferrite core, whose coil is part of an oscillator circuit. When a metallic object enters this field, at some point, the object will absorb enough energy from the field to cause the oscillator to stop oscillating. This signifies that an object is present in a given proximity. See Proximity Sensor.

Industrial Robot

A re-programmable multifunctional manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks (R15.06). The principle components are: one or more arms that can move in several directions, a manipulator, and a computer controller that gives detailed movement instructions.

INFORM

The robot programming language for Yaskawa robots. INFORM language allows the robot user to: instruct the robot to use its basic capabilities to fulfill a defined set of expectations and also to

describe to the robot, through a definition of parameters and conditions, what the expectations are in some given situations or scenarios. In simple terms, the INFORM programming language allows the user to instruct the robot on what to do, when to do it, where to do it and how to do it.

Input Devices

A variety of devices, which allow a human to machine interface. This allows the human to program, control, and simulate the robot. Such devices include programming pendant, computer keyboards, a mouse, joy-sticks, push buttons, operator panel, operator pedestal etc.

Instruction

A line of programming code that causes action from the system controller. See Command position.

Instruction Cycle

The time it takes for a robot controller system's cycle to decode a command or instruction before it is executed. The Instruction Cycle must be analyzed very closely by robotic programmers to enable speedy and proper reaction to varying commands.

Integrate

To fit together different subsystems, such as robots and other automation devices, or at least different versions of subsystems in the same control shell.

Integrator

A company that provides value added services that results in creation of automation solutions by combining a robot and other automation and controls equipment to create an automation solution for end users.

Intelligent Robot

A robot that can be programmed to make performance choices contingent on sensory inputs with little or no help from human intervention. See Robot.

Interference Area

Interference Area is a function that prevents interference between multiple manipulators or the manipulator and peripheral device. The areas can be set up to 64 areas. Three types of methods to use each interference area are as follows: Cubic Interference, Outside of Cubic Area and Axis Interference.

Interpolation

The method by which endpoint paths are created. In general, to specify a motion a few knot points are defined before all the intermediate positions between them are calculated by mathematical interpolation. The interpolation algorithm used therefore has a dramatic effect of the quality of motion.

ISO

International Organization for Standardization

JOB

JOB is the Yaskawa name for a robot program created using Yaskawa's INFORM robot programming language. Typically, a JOB consists of instructions that tell the robot controller what to do and data that the program uses when it is running.

Joint

A part of the manipulator system, which allows a rotation and/or translational degree of freedom of a link of end-effector.

Joint Interpolated Motion

A method of coordinating the movement of the joints, such that all joints arrive at the desired location simultaneously. This method of servo control produces a predictable path regardless of speed and results in the fastest pick and place cycle time for a particular move.

Joint Motion Type

Also known as Point-to-Point Motion, Joint Motion Type is a method of path interpolation that commands the movement of the robot by moving each

joint directly to the commanded position so that all axis arrive to the position at the same time. Although the path is predictable, it will not be linear.

Joint Space

- a. Joint Space (or Joint Coordinates) is just a method of defining the position of the robot in terms of the value of each axis instead of as a TCP position. For example, the Home Position of a robot is often defined in Joint Space as each axis being at 0 degrees.

- b. The set of joint positions.

Joints

The parts of the robot arm which actually bend or move.

Kinematics

The relationship between the motion of the endpoint of a robot and the motion of the joints. For a Cartesian Robot this is a set of simple linear functions (linear tracks that may be arranged in X, Y, Z directions), for a revolute topology (joints that rotate) however, the kinematics are much more complicated involving complicated combinations of trigonometry functions. The kinematics of an arm is normally split into forward and inverse solutions.

Ladle Gripper

An end-effector, which acts as a scoop. It is commonly used to scoop up liquids, transfer it to a mold and pour the liquid into the mold. Common for handling molten metal under hazardous conditions. See End-effector.

Laser

Acronym for Light Amplification by Stimulated Emission of Radiation. A device that produces a coherent monochromatic beam of light which is extremely narrow and focused but still within the visible light spectrum. This is commonly used as a non-contact sensor for robots. Robotic applications include: distance finding, identifying accurate

locations, surface mapping, bar code scanning, cutting, welding etc.

Linear Interpolated Motion

Is a method of path interpolation that commands the movement of the robot by moving each joint in a coordinated motion so that all axis arrive to the position at the same time. The path of the Tool Control Point (TCP) is predictable and will be linear.

Linear Motion Type

Is a method of path interpolation that commands the movement of the robot by moving each joint in a coordinated motion so that all axis arrive to the position at the same time. The path of the Tool Control Point (TCP) is predictable and will be linear.

Link

A rigid part of a manipulator, which connects adjacent joints.

Links

The static material, which connects the joints of an arm together. Thereby a kinematical chain is formed. In a human body, the links are the bones.

Load Cycle Time

A manufacturing or assembly line process term, which describes the complete time to unload the last work piece and load the next one.

Magnetic Detectors

Robot sensors that can sense the presence of ferromagnetic material. Solid-state detectors with appropriate amplification and processing can locate a metal object to a high degree of precision. See Sensor.

Manipulator

A machine or robotic mechanism of which usually consists of a series of segments (jointed or sliding relative to one another) for the purpose of grasping and/or moving objects (pieces or tools), usually in several degrees of freedom. The control of the manipulator may be by an operator, a programmable electronic controller or any

logic system (for example cam device, wired, etc.) (ISO 8373)

Manual Mode

See Teach Mode.

Material Handling

The process by which an industrial robotic arm transfers materials from one place to another.

Material Processing Robot

A robot designed and programmed so that it can machine, cut, form or change the shape, function or properties of materials it handles between the time the materials are first grasped and the time they are released in a manufacturing process.

Mirror Shift Function

With the Mirror Shift Function, a job is converted to the job in which the path is symmetrical to that of the original job. This conversion can be performed for the specified coordinate among the X-Y, X-Z or Y-Z coordinate of the robot coordinates and the user coordinates. The Mirror Shift Function is classified into the following three: the Pulse Mirror Shift Function, the Robot Coordinates Mirror Shift Function and the User Coordinates Mirror Shift Function. (see figure at right)

Mode Switch

As per safety standards, an industrial robot has three distinct modes of operation. These are Teach (also called Manual) and Play (also called Automatic) and Remote. Switching between these modes is performed using a key switch on the teach pendant and is called Mode Switch.

Modularity

The property of flexibility built into a robot and control system by assembling separate units, which can be easily joined to or arranged with other parts or units.

Module

Self-contained component of a package. This component may contain sub-components known as sub-modules.

Motion Axis

The line defining the axis of motion either linear or rotary segment of a manipulator.

Motor

See Servo Motor.

Muting

While testing a robot program, the deactivation of any presence sensing safeguarding devices during the full robot cycle or a portion of the cycle.

Off-line Programming

A programming method where the task program is defined on devices or computers separate from the robot for later input of programming information to the robot. (ISO 8373)b. A means of programming a robot while the robot is functioning. This becomes important in manufacturing and assembly line production due to keeping productivity high while the robot is being programmed for other tasks.

Operator

The person designated to start, monitor and stop the intended productive operation of a robot or robot system. An operator may also interface with a robot for productive purposes. (R15.06)

Optical Encoder

A detection sensor, which measures linear or rotary motion by detecting the movement of markings past a fixed beam of light. This can be used to count revolutions, identify parts, etc.

Optical Proximity Sensors

Robot sensors which measure visible or invisible light reflected from an object to determine distance. Lasers are used for greater accuracy.

Orientation

The angle formed by the major axis of an object relative to a reference axis. It must be defined relative to a three dimensional coordinate system. Angular position of an object with respect to the robot's reference system. See Roll, Pitch and Yaw.

Palletizing

The process of stacking packages (i.e., boxes, bags, containers, etc.) in an organized fashion on a pallet.

PAM Function – Position Adjustment by Manual

Position Adjustment by Manual allows position adjustment by simple operations while observing the motion of the manipulator, and without stopping the manipulator. Positions can be adjusted in both teach mode and play mode.

Parallel Shift Function

Parallel Shift refers to the shifting of an object from a fixed position in such a way that all points within the object move an equal distance. In the model for Parallel Shift shown in the following, the shift value can be defined as the distance L (three dimensional coordinate displacement). The Parallel Shift Function is relevant to the actual operation of the manipulator because it can be used to reduce the amount of work involved in teaching by shifting a taught path (or position). In the example in the figure below, the taught position A is shifted in increments of the distance L (this is actually a three dimensional XYZ displacement that can be recognized by the robot).

Path

The continuous locus of positions (or points in three dimensional space) traversed by the tool center point and described in a specified coordinate system. (R15.05-2)

Payload – Maximum

The maximum mass that the robot can manipulate at a specified speed, acceleration/deceleration, center of gravity location (offset), and repeatability under continuous operation over a specified working space. Maximum payload is specified in kilograms. (R15.05-2)

Pendant [Teach Pendant]

A hand-held input device, linked to the control system with which a robot can be

programmed or moved. (ISO 8373) This enables the human operator to stand in the most favorable position to observe, control and record the desired movements in the robot's memory.

Pendant Teaching

The mapping and recording of the position and orientation of a robot and/or manipulator system as the robot is manually moved in increments from an initial state along a path to a final goal state. The position and orientation of each critical point (joints, robot base, etc.) is recorded and stored in a database for each taught position the robot passes through on its path toward its final goal. The robot may now repeat the path on its own by following the path stored in the database.

Performance Level d (PLd)

ISO Performance Level (PL) “d” means that the average probability of dangerous failure per hour of the safety related parts of the control system falls within $\geq 10^{-7}$ to $< 10^{-6}$. Additionally, other factors such as proper installation, maintenance and protection against environmental factors also apply. This is the minimum performance level specified in ISO 10218-2 section 5.2.2, unless a risk assessment would allow a lower value to be used.

Performance Level e (PLe)

ISO Performance Level (PL) “e” means that the average probability of dangerous failure per hour of the safety related parts of the control system falls within $\geq 10^{-8}$ to $< 10^{-7}$. Additionally, other factors such as proper installation, maintenance and protection against environmental factors also apply.

Pick and Place Cycle

The amount of time it takes for a manipulator to pick up an object and place it in a desired location, then return to its rest position. This includes time during the acceleration and deceleration phases of a particular task. The robot movement is controlled from one point location in space to another in a Point-to-Point (PTP) motion system. Each point is programmed

into the robot's control memory and then played back during the work cycle.

Pick-and-Place Task

A repetitive part transfer task composed of a picking action followed by a placing action.

Pinch Points

A pinch point is any point at which it is possible for a person or part of a person's body to be caught between moving parts of a machine, or between the moving and stationary parts of a machine, or between material and any part of the machine. A pinch point does not have to cause injury to a limb or body part, although it might cause injury – it only has to trap or pinch the person to prevent them from escaping or removing the trapped part from the pinch point.

Pitch

Rotation of the end-effector in a vertical plane around the end of the robot manipulator arm.

See Roll and Yaw.

Play Mode

After a robot is programmed in Teach Mode, the robot controller can be switched to Play Mode to execute the robot program. In Play Mode, the robot program is played back. This is the mode in which robots are used in production.

Playback Operation

Playback is the operation by which the taught job is played back. This function is used to decide where to resume the playback on the start operation after suspending the playback and moving the cursor or selecting other jobs. 0: Starts operation where the cursor is located in the job displayed at the moment. 1: The playback continuation window appears. Select "YES" and the playback resumes where the cursor has been located when the playback suspended. If "NO" is selected, the playback resumes where the cursor is located in the job displayed at the moment. Modes Switch on the programming pendant: PLAY – job is

started up by [START] on the programming pendant and REMOTE job is started by a peripheral device (external start input).

Point-to-Point

Manipulator motion in which a limited number of points along a projected path of motion is specified. The manipulator moves from point to point rather than a continuous smooth path.

Pose

Alternative term for robot configuration, which describes the linear and angular position. The linear position includes the azimuth, elevation and range of the object. The angular position includes the roll, pitch and yaw of the object. See Roll, Pitch and Yaw.

Position

The definition of an object's location in 3D space, usually defined by a 3D coordinate system using X, Y and Z coordinates.

Position Level

The position level is the degree of approximation of the manipulator to a taught position. The position level can be added to move instructions MOVJ (joint interpolation) and MOVL (linear interpolation). If the position level is not set, the precision depends on the operation speed. Setting an appropriate level moves the manipulator in a path suitable to circumferential conditions and the workpiece. (see figure at right)

Position Variables

Position Variables are used in a robot program (JOB) to define a location in 3D space, usually defined by a 3D coordinate system using X, Y and Z coordinates. As it is a variable, the value can change depending on conditions or on information passed to the JOB.

Power and Force Limiting (PFL)

Collaborative feature that allows both the operator and robot to work in proximity to one another by ensuring the robot will slow down and stop before a contact

situation occurs. In order for this feature to be safely implemented, functional safety and additional detection hardware must be used. A risk assessment shall be used to determine if any additional safeguarding is necessary to mitigate risks within the robot system.

Presence-sensing Safeguarding Device

A device designed, constructed and installed to create a sensing field to detect an intrusion into such field by people, robots or objects. See Sensor.

Programmable Logical Controller (PLC)

A solid-state control system, which has a user programmable memory for storage of instructions to implement specific functions such as: I/O control logic, timing, counting arithmetic and data manipulation. A PLC consists of a central processor, input/output interface, memory and programming device, which typically uses relay equivalent symbols. The PLC is purposely designed as an industrial control system, which may perform functions equivalent to a relay panel or a wired solid-state logic control system, and may be integrated into the robot control system.

Programmable Robot

A feature that allows a robot to be instructed to perform a sequence of steps and then to perform this sequence in a repetitive manner. It can then be reprogrammed to perform a different sequence of steps if desired.

Proximity Sensor

A non-contact sensing device used to sense when objects are a short distance away, and it can determine the distance of the object. Several types include: radio frequency, magnetic bridge, ultrasonic and photoelectric. Commonly used for: high speed counting, sensing metal objects, level control, reading coding marks and limit switches. See Inductive Sensor.

Pulse Coordinates

Yaskawa robots define robot joint axes position in degrees for revolute joints.

Pulse is also another way to specify robot joint position, and it does so in robot motor encoder pulse counts.

Quality Assurance (QA)

Describes the methods, policies and procedures necessary to conduct quality assurance testing during design, manufacturing and deliver phases of creating, reprogramming, or maintaining robots.

Reach: The volume of space (envelope), which a robot's end-effector can reach in at least one orientation.

Quasi-static Clamping

A type of contact between a person and part of a robot system where the body part can be clamped between the moving part of the robot system & another fixed or moving part of the robot cell

Reach

The volume of space (envelope), which a robot's end-effector can reach in at least one orientation.

Real-time System

A computer system in which the computer is required to perform its tasks within the time restraints of some process simultaneously with the system it is assisting. The computer processes system data (input) from the sensors for the purpose of monitoring and computing system control parameters (outputs) required for the correct operation of a system or process. The computer is required to do its work fast enough to keep pace with an operator interacting with it through a terminal device (such as a screen or keyboard). The operator interacting with the computer has access, retrieval and storage capability through a database management system. System access allows the operator to intervene and alter the system's operation.

Record-playback Robot

A manipulator for which the critical points along desired trajectories are stored in sequence by recording the actual values of the joint-position encoders of the robot as

it is moved under operational control. To perform the task, these points are played back to the robot's servo-system. See Servo-system.

Rectangular-Coordinate Robot

A robot whose manipulator arm moves in linear motions along a set of Cartesian or rectangular axis in X, Y and Z directions. The shape of the work envelope forms a rectangular figure. See Work Envelope.

Reliability

The probability or percentage of time that a device will function without failure over a specified time period or amount of usage (R15.02). Also called: the robot's uptime or the Mean Time Between Failure (MTBF).

Remanufacture

To upgrade or modify robots to the revised specifications of the manufacturer. (R15.06)

Remote Mode

Remote Mode is a type of Play Mode where the automatic execution of robot program is initiated from an external device (not the teach pendant). During this mode, the use of the teach pendant is disabled.

Repeatability

A measure of how close an arm can repeatedly obtain a taught position. For instance: once a manipulator is manually placed in a particular location and this location is resolved by the robot, the repeatability specifies how accurately the manipulator can return to that exact location. The degree of resolution within the robot control system determines the repeatability. In general, an arm's repeatability can never be better than its resolution. See Teach and Accuracy.

Resolution

The amount of robot joint motion required for the position sensing to change by one count. Although the resolution of each joint feedback sensor is normally constant, the resolution of the endpoint in world

coordinates is not constant for revolute arms, due to the non-linearity of the arm's kinematics.

Revolute Joint

The joints of a robot, which are capable of rotary motion.

Risk Assessment

The process of evaluating the intended use of a machine or system for foreseeable hazards and then determining the level of risk involved for the tasks identified.

Risk Mitigation

A secondary step in the risk assessment process that involves reducing the level of risk for the identified tasks, by applying risk reduction measures in order to eliminate or mitigate the hazards.

Robot

A re-programmable, multifunctional manipulator designed to move material, parts, tools or specified devices through variable programmed motions for the performance of a variety of tasks. Common elements which make up a robot are: controller, manipulator and end-effector. See Manipulator, Controller and End-effector.

Robot Coordinate System

The Robot Coordinate System is defined in the base axis of a Robot, and points in the Robot Coordinate System will be relative to the base of the robot. Note that by default the Base Coordinate System and Robot Coordinate System are the same. (see figure at right)

Robot Integrator

See Integrator.

Robot Programming Language

An interface between a human user and a robot, which relates human commands to the robot.

Robot Range Limit Monitoring

Monitors the manipulator arm or its tool to be in the designated safety area

Robot Simulation

A method for emulating and predicting the behavior and the operation of a robotic system based on the model (i.e., computer graphics) of the physical system. (R15.07)

Roll

Rotation of the robot end effector in a plane perpendicular to the end of the manipulator arm. See Pitch and Yaw.

Rotary Joint

A joint which twists, swings or bends about an axis.

Rotary Vector Drive (RV)

A brand name for a speed reduction device that converts high speed low torque to low speed high torque, usually used on the major (larger) axis . See Cyclo Drive and Harmonic Drive.

Rotational Motion

A joint which twists, swings or bends about an axis. An example of this is the elbow of a human arm.

Safeguard

A barrier guard, device or safety procedure designed for the protection of personnel.

Safety Logic Circuit

The safety logic circuit monitors safety critical external devices such as the light curtains and FSU generated signals. The safety logic circuit is programmed via an intuitive user interface that is supported on the Yaskawa programming pendant. It enables to set up the logical operations, such as stopping the manipulator or outputting a signal if the servos are on.

Safety Monitored Stop

Collaborative feature designed to allow safe human-robot interaction. Only when robot motion ceases can the human safety enter the collaborative workspace. Servos can remain energized in accordance with a category 2 stop in accordance with ISO 10218-1:2011, 5.4. A risk assessment shall be used to determine if any additional safeguarding is necessary to mitigate risks within the robot system.

SCARA Robot

A cylindrical robot, having two parallel rotary joints (horizontally articulated) and provides compliance in one selected plane. (ISO 8373) Note: SCARA derives from Selectively Compliant Arm for Robotic Assembly

Second Home Position

Apart from the “home position” of the manipulator, the second home position can be set up as a check point for absolute data. The initial value of the second home position is the home position (where all axes are at pulse 0). The second home position can be changed.

Security Mode

Levels of operator modes on Yaskawa robot controllers, include: Operation Mode, Edit Mode, Management Mode, Safety Mode and One Time Management mode.

Sensor

Instruments used as input devices for robots, which enable it to determine aspects regarding the robot's environment, as well as the robot's own positioning. Sensors respond to physical stimuli (such as heat, light, sound, pressure, magnetism and motion), and they transmit the resulting signal or data for providing a measurement, operating a control or both. (R15.06)

Sensory Feedback

Variable data measured by sensors and relayed to the controller in a Closed-loop System. If the controller receives feedback that lies outside an acceptable range, then an error has occurred. The controller sends an error signal to the robot. The robot makes the necessary adjustments in accordance with the error signal.

Servo Control

The process by which the control system of the robot checks if the attained pose of the robot corresponds to the pose specified by the motion planning with required performance and safety criteria. (ISO 8373)

Servo Motor

An electrical power mechanism used to effect motion or maintains position of the robot (for example, a motor which converts electrical energy to effect motion of the robot) (R15.07). The motor responds to a signal received from the control system and often incorporates an encoder to provide feedback to the control loop.

Servo Pack

An alternating, current electrical power mechanism that is controlled through logic to convert electrical supply power that is in a sine wave form to a Pulse Width Modulated (PWM) square form, delivered to the motors for motor control: speed, direction, acceleration, deceleration and braking control.

Servo-controlled Robot

The control of a robot through the use of a Closed-loop Servo-system, in which the position of the robot axis is measured by feedback devices and is stored in the controller's memory. See Closed-loop System and Servo-system.

Servo-system

A system in which the controller issues commands to the motors, the motors drive the arm, and an encoder sensor measures the motor rotary motions and signals the amount of the motion back to the controller. This process is continued many times per second until the arm is repositioned to the point requested. See Servo-controlled Robot

Shock Detection Function

Shock detection is a function supported by the Yaskawa robot controller that reduces the impact of a robot collision by stopping the manipulator without any external sensor when the tool or the manipulator collide with a peripheral device.

Shoulder

The first or second axis of a robot is sometimes referred to as a shoulder axis as it somewhat resembles a human shoulder. This is often used in describing humanoid

or dual-arm systems such as the Yaskawa Motoman® SDA10D.

SIL

See Safety Integrity Level

Simulation

A graphical computer program that represents the robot and its environment, which emulates the robot's behavior during a simulated run of the robot. This is used to determine a robot's behavior in certain situations, before actually commanding the robot to perform such tasks. Simulation items to consider are: the 3D modeling of the environment, kinematics emulation, path-planning emulation and simulation of sensors.

Singularity

A configuration where two joints of the robot arm become co-axial (aligned along a common axis). In a singular configuration, smooth path following is normally impossible and the robot may lose control. The term originates from the behavior of the Jacobian matrix, which becomes singular (i.e., has no inverse) in these configurations.

SLURBT

SLURBT are terms that Yaskawa Motoman uses to describe each axis of the robot for convenience. The definition of each value is as follows:

S – Swing or Swivel

L – Lower Arm

U – Upper Arm

R – Rotate

B – Bend

T – Twist

Softlimit Setting Function

The Softlimit Setting Function is a function to set the axis travel limit range of the manipulator motion in software.

Speed and Separation Monitoring

Collaborative feature that allows both the operator and robot to work in proximity to one another by ensuring the robot will slow down and stop before a contact situation occurs. In order for this feature to

be safely implemented, functional safety and additional detection hardware must be used. A risk assessment shall be used to determine if any additional safeguarding is necessary to mitigate risks within the robot system.

Spline

A smooth, continuous function used to approximate a set of functions that are uniquely defined on a set of sub-intervals. The approximating function and the set of functions being approximated intersect at a sufficient number of points to insure a high degree of accuracy in the approximation. The purpose for the smooth function is to allow a robot manipulator to complete a task without jerky motion.

Spline Motion Type

A calculated path that the robot executes that may be parabolic in shape. A spline motion may also accomplish a free form curve with mixtures of circular and parabolic shapes.

System Integrator

See Integrator.

Teach

To program a manipulator arm by manually guiding it through a series of motions and recording the position in the robot controller memory for playback.

Teach Lock

While the Teach Lock is set, the mode of operation is tied to the Teach Mode and the machines cannot be played back using either [START] or external input. For safety purposes, always set the mode switch to "TEACH" before beginning to teach.

Teach Mode

A robot controller mode in which a robot manipulator is programmed by manually guiding it through a series of motions and recording the position in the robot controller memory for playback. Industrial robots that do not have an active Power and Force Limiting Function require the

use of a Three Position Enable Switch in Teach Mode.

Teach Pendant

A handheld control box, which is used by an operator to remotely guide a robot through the motions of its tasks. The motions are recorded by the robot control system for future playback. Modern industrial robots come with programming pendants, which not only allow robot teaching, but also support full feature robot programming and safety user interface.

Teaching Window

Teaching Window is a user interface screen on the programming pendant. This window contains the JOB CONTENT window and teaching is conducted within this window. The JOB CONTENT window contains the following items: line numbers, cursor, instructions, additional items, comments, etc.

Through-beam

An object detection system used within a robot's imaging sensor system. A finely focused beam of light is mounted at one end and a detector at the other. When the beam of light is broken, an object is sensed.

Time Measuring Function

Time measuring function measures the execution time for the specified section in the job or the signal output time of the specified signal.

Tool

A term used loosely to define a working apparatus mounted to the end of the robot arm, such as a hand, gripper, welding torch, screw driver, etc. See Arm, Gripper and End-effector.

Tool & arm Interference

In a system with one controller and multiple manipulators, the Tool & Arm Interference Check Function can be used to detect possible interference to avoid collision during operation. The following three patterns can be checked:

- Arm against arm

- Arm against tool
- Tool against tool

Interference is checked by using a cylinder that is slightly larger than the arm or tool. A sphere is placed on both ends of the cylinder. If the cylinder and spheres of one manipulator have any contact with those of the other while moving, the manipulators stop because interference was detected.

Tool Center Point (TCP)

The Tool Center Point (TCP) defines the tip of the current tool as defined relative to the tool flange. For example, for a welding robot, the TCP will generally be defined at the tip of the welding gun. After defining and configuring the TCP, the robot motion will be defined relative to this frame (i.e., rotation in the Rx direction would cause rotation around the X-axis and positions will be taught in this frame.

Tool Coordinates

When the tool attached to the robot moves, so does its tool coordinate system in reference to a fixed coordinate system, for example, world coordinates. In general, the tool coordinates do not align with the world XYZ coordinates.

Tool Frame

A coordinate system attached to the end-effector of a robot (relative to the base frame).

Touch Sensor

Sensing device, sometimes used with the robot's hand or gripper, which senses physical contact with an object, thus giving the robot an artificial sense of touch. The sensors respond to contact forces that arise between themselves and solid objects.

Trajectory Generation (Calculation)

The computation of motion functions that allow the movement of joints in a smooth controlled manner.

Transducer

A device that converts energy from one form to another. Generally, a device that converts an input signal into an output

signal of a different form. It can also be thought of as a device which converts static signals detected in the environment (such as pressure) into an electrical signal that is sent to a robot's control system.

Uptime

A period of time in which a robot or production line is operating or available to operate, as opposed to downtime.

User Coordinate Setting

User coordinates are defined by three points that have been taught to the manipulator through axis operations. These three defining points are ORG, XX, and XY, as shown in the diagram below. These three points of positional data are registered in a user coordinate file. ORG is the home position, and XX is a point on the X-axis. XY is a point on the Y-axis side of the user coordinates that has been taught, and the directions of Y- and Z-axes are determined by point XY.

User Coordinate System

The User Coordinate System is any reference point that a user has defined for their application. This is often attached to an object such as a pallet and allows a user to teach points relative to this object. For example, a set of position could be taught relative to a User Coordinate System attached to a pallet and then easily transferred to a different User Coordinate System on another pallet. This allows for positions to be reused efficiently. See also, User Coordinate Setting

Vacuum Cup Han

An end-effector for a robot arm which is used to grasp light to moderate weight objects, using suction, for manipulation. Such objects may include glass, plastic; etc. Commonly used because of its virtues of reduced object slide slipping while within the grasp of the vacuum cup. See End-effector.

Vision Guided

Control system where the trajectory of the robot is altered in response to input from a vision system.

Vision Sensor

A sensor that identifies the shape, location, orientation, or dimensions of an object through visual feedback, such as a television camera.

Work Envelope

The set of all points which a manipulator can reach without intrusion. Sometimes the shape of the work space, and the position of the manipulator itself can restrict the work envelope.

Work Envelope (Space)

The volume of space within which the robot can perform given tasks.

Work Home Position

The Work Home Position is a reference point for manipulator operations. It prevents interference with peripheral device by ensuring that the manipulator is always within a set range as a precondition for operations such as starting the line. The manipulator can be moved to the set Work Home Position by operation from the programming pendant, or by signal input from an external device. When the manipulator is in the vicinity of the Work Home Position, the Work Home Position signal turns ON.

Work Piece

Any part which is being worked, refined or manufactured prior to its becoming a finished product.

Workspace

The volume of space within which the robot can perform given tasks.

World Coordinates

A reference coordinate system in which the manipulator arm moves in linear motions along a set of Cartesian or rectangular axes in X, Y, and Z directions. The shape of the work envelope forms a rectangular figure. See Rectangular Coordinates.

World Model

A three dimensional representation of the robot's work environment, including

objects and their position and orientation in this environment, which is stored in robot memory. As objects are sensed within the environment the robot's controller system continually updates the World Model. Robots use this World Model to aid in determining its actions in order to complete given tasks.

Wrist

A set of rotary joints between the arm and the robot end-effector that allow the end-effector to be oriented to the work-piece. In most cases the wrist can have degrees of freedom which enable it to grasp an object with roll, pitch, and yaw orientation. See Arm, End-effector, Roll, Pitch, Yaw and Work Piece.

Wrist [Secondary Axis]

An interconnected set of links and powered joints between the arm and end-effector, which supports, positions and orientates the end effector.

Yaw

Rotation of the end-effector in a horizontal plane around the end of the manipulator arm. Side to side motion at an axis. See Roll and Pitch

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.