Appropriate Deployment of Robots can aid Nigerian Manufacturing Industries to compete favourably in the global Marketplace

[1]Prof Oliver E., Osuagwu [1] ; Oliver, Obilor Osuagwu [2]

[1] *Department of Computer Science/ Imo State University, Owerri, Imo State, Nigeria*

[2] *Department of Electrical Electronics Engineering/ Imo State University, Owerri, Imo State, Nigeria*
+234 816 629 7586

Abstract:
This paper is geared towards raising awareness of the need to develop skills in robot design, production, and programming. The robot revolution has positively impacted industries worldwide, particularly in cost reduction, product quality, and competitive pricing. Corporations that embrace automation are able to compete effectively in international markets. In Africa, however, the reverse is the case. For Africans to remain relevant in the comity of nations, it is imperative to strengthen knowledge workers by equipping them with new skills, particularly in intelligent machine and robot design, as well as the associated Artificial Intelligence (AI) programming components.
This paper contends that the absence of innovation in science and technology will leave Africa a century behind the rest of the world if concerted efforts are not made by African governments to avert this envisaged catastrophe through political will and effective, pragmatic national planning. The paper addresses the definition of a robot, its key components, local content input, integration, and programming, as well as its contribution to national economic growth. There is therefore an urgent need for active government policy thrust to support aggressive skill acquisition in machine and robot design, fabrication, and programming, utilizing available local material content within the nation's Departments of Mechanical Engineering, Electronics and Computer Engineering, Mathematics, and Computing Sciences.
Conclusion: *Learning the skill to program robots will enable African industries to embrace robotics and foster competition in precision, quality, and pricing of manufactured goods. Africa cannot achieve sustainable development without this caliber of knowledge workers, as true progress is driven by innovation in science and technology.*
It is therefore recommended that African governments actively promote the design, manufacture, and programming of intelligent robots for industrial use. Such initiatives will generate a snowball effect across industries, enhancing market competition, product quality, pricing, and packaging. This paper hopes to inspire pragmatic automation in African industries, ensuring the continent's relevance in the global marketplace.
Keywords: *Robotics in Africa, Intelligent machines, Artificial Intelligence (AI) programming, Industrial automation, Manufacturing competitiveness, Innovation in science and technology, Robot design and fabrication, Knowledge workers in Africa, Global marketplace integration, Smart manufacturing, National economic growth, Local content development, Pragmatic automation, Government policy on robotics, Skill acquisition in engineering*

1.0 Introduction

1.1 Application of Robots in the Industry

A robot is a multi-function, programmable machine that imitates the actions or appearance of an intelligent creature—usually a human. Engineering Innovation [1] defines a robot as *"a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks."*

For a machine to qualify as a robot, it must be able to:

i. **Sense and perceive**: gather information from its surroundings

ii. **Carry out different tasks**: locomotion or manipulation, performing physical actions

iii. **Move or manipulate objects**

iv. **Be re-programmable**: capable of performing different functions

v. **Function autonomously** and/or interact with human beings

1.2 Attributes of Intelligent Robots

Robots described as intelligent ought to possess the following attributes:

i. **Autonomy**: ability to achieve task objectives without human input, making and executing decisions based on sensor information

ii. **Intuitive Human-Robot Interfaces**: commands should be natural and require minimal training, especially in smart home applications

iii. **Adaptation**: Ability to adjust to changes in the environment. Thus, robot systems operating in intelligent environments require capabilities such as:

i. Autonomous control systems

ii. Simple and natural human-robot interfaces

iii. Adaptive and learning capabilities

iv. Safety maintenance during operation

1.3 Major Parts of a Robot

Robots generally consist of three major parts:

1. **Mechanical Base** – The frame on which a robot sits and operates, usually mechanical or plastic.
2. **Actuation Mechanism** – A device designed to move a control member in automatic or remote-control systems, serving as an auxiliary drive in servomechanisms and vehicular steering gear. It alters energy or material flow to an object, ensuring minimal deviations from prescribed values. Typically, it is powered by external sources, as direct control via relays or transducers is insufficient.
3. **Control** – Robot control refers to the way in which the sensing and action of a robot are coordinated. There are infinitely many possible robot programs, but they all fall along a well-defined spectrum of control. Along this spectrum, four basic practical approaches are being used today:

i. Deliberative control

ii. Reactive control

iii. Hybrid control

iv. **Behavior-based control:** Robot intelligence is made possible through specialized robot programming languages.

What makes a robot intelligent is the software driver written by AI software engineers to control Kinematics and dynamic actions such as Simple and Compound joints, Mobility, Work areas and Coordinates

1.4 Robotics as a Discipline

Robotics is the science and technology of building robots, and comprises multiple fields shown in **Fig 1**, including:

i. Control Theory

ii. Vision systems

iii. Artificial intelligence

iv. Mechanical engineering

v. Information technology

vi. Electrical engineering

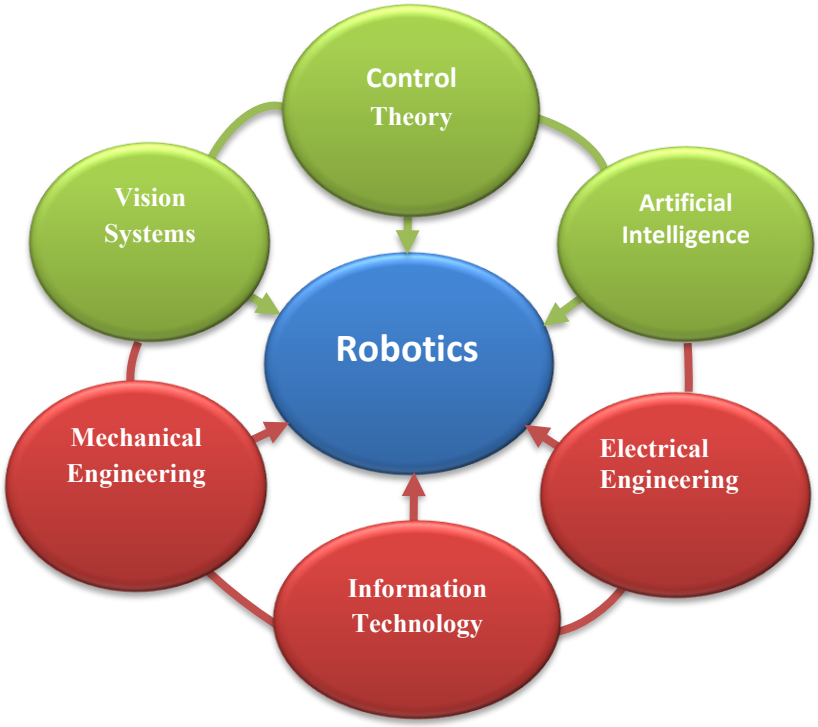


Fig. 1: ***Composition of Robot Knowledge Areas***

2.0 Robot Intelligence and Computer Programs

What makes a robot intelligent is the software driver written by AI software engineers to control kinematics and dynamic actions such as simple and compound joints, mobility, work areas, and coordinates.

3.0 Types of Robots

3.1 Many types of robots exist today. The progress of technology allows us now to build an intelligent robot with perception and manipulation under $1,000. There are several kits for even high school robotics.

3.2 Robotics allows us to explain ideas of control, software, sensors, motors, and other effectors, image processing, pattern recognition, machine learning, and Artificial Intelligence.

3.3 Some kinds of robots in the marketplace today include:

i. **Driving Robots:** Uses differential steering, synchro-drive, and tracked vehicles. They are autonomous cars, also known as driverless cars, self-driving cars, or robot cars. It is an autonomous vehicle capable of fulfilling human transportation.

ii. **Omni-Directional Robots:** Deploys Mecanum wheels. This type of robot can have a significant advantage over conventional robots. Each robot can drive in any direction, i.e., forward/backward, sideways, or at an angle.

iii. Flying Robots: Sensors and structure for surveillance and detection. Equipped with appropriate sensors, the FR SWAN X1 collects and transmits real-time images useful for decision-making.

iv. Balancing Robots: Deploys an inverted pendulum. They are simple robots that use a simple switch as a sensor and stand on only two wheels with an inverted pendulum mechanism. When the robot is going to fall, the motor starts and moves the robot in the direction it is going to fall, so the motor torque about the center of gravity is higher than the motor torque required to keep the robot balanced.

v. Walking Robots: Six-legged walking robots, bipedal android robots. A mobile robot is an automatic machine that is capable of movement in any given environment. Mobile robots have the capability to move around in their environment and are not fixed to one physical location.

4.0 Knowledge Acquisition

Modern robots need knowledge acquisition capabilities via the deployment of the following related technologies:

i. Sensors

ii. Cameras

iii. Radar

iv. Tactile sensors

v. Infrared, etc.

Decision-making systems are mainly Artificial Intelligence programs.

5.0 History of World Robot Development

S/N	Year	History of World Robot Development
1	200 BC	Greek Development for Theatre Automation (*Greeks created automated theatre mechanisms, achieving lifelike stage effects through mechanical engineering*.)
2	1500	Leonardo da Vinci (*Leonardo da Vinci designed mechanical automata, including a humanoid knight)*
3	1650-1774	Robot dolls (scribe 1774) *(Early automata were developed that could write, demonstrating mechanical precision and lifelike motion)*
4	1968	"Artifical Intelligence' era, shakey, 30min./m *(Shakey the Robot: First mobile robot controlled by Artificial Intelligence, achieving autonomous navigation and decision-making*)
5	1980	Transport Units in Factories *(Robots began moving materials in factories, improving efficiency and reducing human labor)*
6	1985	"Showboats" (Wasubot Piano-Player,etc.): ***(ShowBots & MIT Leg Lab:*** *Entertainment robots demonstrated interactive automation, while MIT advanced robotic locomotion research)*
7	1989	TRC Helpmate: *(Introduced as a mobile service robot, capable of assisting in hospitals and workplaces.)*
8	1990	Service Robots: *(Cleaning and assistance robots emerged, marking the rise of practical household and industrial automation)*
9	1995	Autonomous sojourner ("Low cost"): *(Autonomous Automobiles: Early prototypes of self-driving cars demonstrated robotic navigation in real-world environments)*

10	1997	Mars Rover Sojourner *(NASA's rover achieved autonomous exploration on Mars, proving robotics in space missions)*
11	2002-2020	Advanced Humanoids: *(Robots like ASIMO and Atlas achieved walking, balancing, and complex tasks, driven by AI and machine learning)*

5.1 Robot Applications: The African Imperative

Today, besides the typical use of industrial robots in the manufacturing industry, new application areas for robots have been appearing during the past few years. In these areas, service robots perform general-purpose activities without requiring repeated high-speed movements or other requirements of industrial robot applications.

Thus, in modern economies, the developed West has moved production plants demanding a huge labour force to developing countries where labour is cheap in order to compete in the international markets.

Alternatively, they outsource such jobs demanding manual labour if the cost-benefit analysis justifies this option. This becomes inevitable, particularly where the task involved cannot deploy robots to reduce the cost of production. Consequently, the cost of products coming from the West has the capacity to compete in terms of price and quality in the world marketplace.

For African industries to compete like those in the West, an automation culture has to be imbibed. Most robot fabrication requires the services of Mechanical Engineers, Electrical Engineers, Computer/AI programmers, Information Technology Professionals, etc. These high-caliber knowledge workers need advanced skills in AI and robotics programming for African industries to be able to benefit from this cost-reduction technology. The governments in Africa need to invest in this technology now.

5.2 Benefits of Robot Deployment

Robots are excellent for application in risky tasks where machines, instead of human beings, can be sacrificed. The benefits that may accrue to an industrialist deploying robots include [2]:

Quality:
Industrial automated robots have the capacity to dramatically improve product quality. Applications are performed with precision and high repeatability every time. This level of consistency can be hard to achieve any other way.

Production:
With robots, throughput speeds increase, which directly impacts on production. Because an automated robot has the ability to work at a constant speed without pausing for breaks, sleep, or vacations, it has the potential to produce more than a human worker.

Safety:
Robots increase workplace safety. Workers are moved to supervisory roles where they no longer have to perform dangerous applications in hazardous settings.

Savings:
Improved worker safety leads to financial savings. There are fewer healthcare and insurance concerns for employers. Automated robots also offer untiring performance which saves valuable time. Their movements are always exact, minimizing material waste.

These benefits are bound to help African industries compete in the international marketplace.

5.3 Designing and Building an Indigenous Robot

Step 1: The first step is to determine what your robot should do (i.e., what is its purpose in life). Robots can be used in almost any situation and are primarily intended to help humans in some way.

Step 2: Decide on the type of robot you want to build and what the robot would look like to best suit your mission.

Step 3: Choose the right actuators (e.g., motors) for your chosen robot.

Step 4: Choose your microcontroller. A microcontroller is a computing device capable of executing a program (i.e., a sequence of instructions) and is often referred to as the "brain" or "control center" in a robot since it is usually responsible for all computations, decision making, and communications, as shown next.

5.4 Robot Programming 1

Computer programming consists of writing lines of code in a language that a computer will understand to solve a problem. Some computer languages include: Java, C++, and Python. Python is said to be a programming language that lets you work quickly and integrate systems more effectively. Calculations are simple with Python, and expression syntax is straightforward [4] [5].

Computer science or programming can solve problems ranging from adding numbers together to operating a rover on the Mars planet. Python is one of the key languages for robotic programming. It is easier to deploy Python to talk to robots and have them make noises, move, take pictures, draw, and explore their environment.

For instance, the Scribbler Robot deploys two commands to tell the robot to make noise: **beep** and **speak**.

i. **beep()** takes in information about the number of seconds and frequency of a note. Example: beep(1,440) plays a one-second note.

ii. **speak()** takes in a message in quotes. Example: speak("Hello").

5.5 Microcontroller

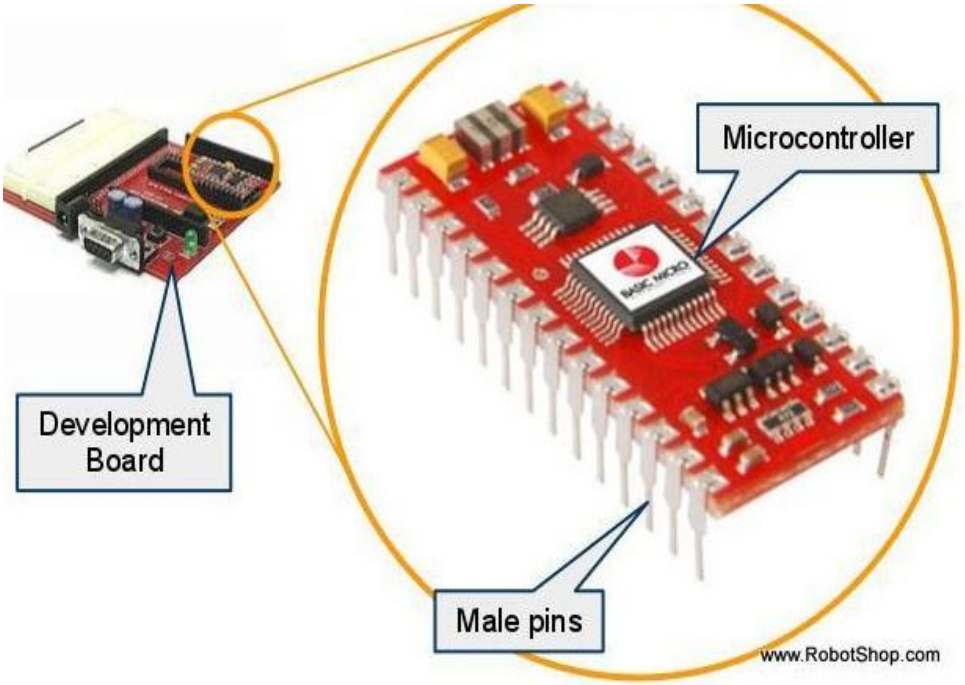


Fig. 2: *Microcontroller (Source: www.Robotshop.com)*

In more advanced robots, especially those that involve complex computing and vision algorithms, the microcontroller is often replaced with a standard computer.

A desktop computer includes a motherboard, a processor, a main storage device (such as a hard drive), video processing (on-board or external), RAM, and, of course, peripherals such as a monitor, keyboard, mouse, etc. This type of system is usually more expensive, physically larger, and more power hungry.

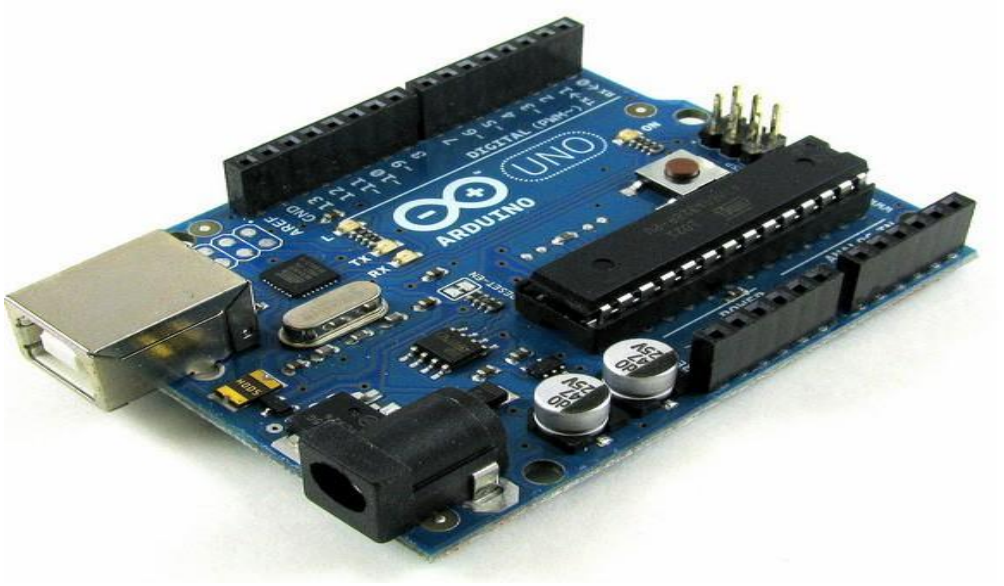


Fig. 3: *An Arduino microcontroller – Arduino Uno*

5.6 What is an Actuator?

An "actuator" can be defined as a device that converts energy (in robotics, that energy tends to be electrical) into physical motion. The vast majority of actuators produce either rotational or linear motion. For instance, a "**DC** motor" is therefore a type of actuator.

Choosing the right actuators for your robot requires an understanding of what actuators are available, some imagination, and a bit of math and physics.

Actuators can be Pneumatic and Hydraulic or AC Motor, DC Motor, Geared DC Motors, R/C Servo Motors, Industrial Servo Motors, Stepper Motors, DC Linear Actuator, and Solenoids.

In choosing an actuator, ask yourself:

i. Will the motor be used to lift or turn a heavy weight?
ii. Is the range of motion limited to 180 degrees?
iii. Does the angle need to be very precise?
iv. Is the motion in a straight line?

A Motor Controller is an electronic device (usually comes in the shape of a bare circuit board without enclosure) that acts as an intermediate device between a microcontroller, a power supply or batteries, and the motors.

Types of Actuators include:

i. Pneumatic and Hydraulic

ii. AC Motor

iii. DC Motor

iv. Geared DC Motors

v. R/C Servo Motors

vi. Industrial Servo Motors

vii. Stepper Motors

viii. DC Linear Actuator

ix. Solenoids

Selection considerations:

i. Will the motor be used to lift or turn a heavy weight?

ii. Is the range of motion limited to 180 degrees?

iii. Does the angle need to be very precise?

iv. Is the motion in a straight line?

A motor controller is an electronic device (usually comes in the shape of a bare circuit board without enclosure) that acts as an intermediate device between a microcontroller, a power supply or batteries, and the motors.

5.7 Types of Motor Controllers

i. Brushed DC motor controllers: Used with brushed DC, DC gear motors, and many linear actuators.

ii. Brushless DC motor controllers: Used with brushless DC motors.

iii. Servo Motor Controllers: Used for hobby servo motors.

iv. Stepper Motor Controllers: Used with unipolar or bipolar stepper motors, depending on their kind.

5.8 Controlling the Robot

The robot can be controlled using:

5.8.1 Direct Wired Control

The easiest way to control a vehicle is with a handheld controller physically connected to the vehicle using a cable (i.e., a tether). Toggle switches, knobs, levers, joysticks, and buttons on this controller allow the user to control the vehicle without the need to incorporate complex electronics. In this situation, the motors and a power source can be connected directly with a switch in order to control its forward/backwards rotation. Such vehicles usually have no intelligence and are considered to be more "remote-controlled machines" than "robots."

5.8.2 Wired Computer Control

The next step is to incorporate a microcontroller into the vehicle, but continue to use a tether. Connecting the microcontroller to one of your computer's I/O ports (e.g., a USB port) allows you to control its actions using a keyboard (or keypad), joystick, or other peripheral device.

Adding a microcontroller to a project may also require you to program how the robot reacts to the input. Instead of using a laptop or desktop computer, notebooks are often a desirable choice because of their low price, small size, and low weight.

5.8.3 Wireless – Infrared

Infrared transmitters and receivers cut the cables connecting the robot to the operator. This is usually a milestone for beginners. Infrared control requires "line of sight" in order to function; the receiver must be able to "see" the transmitter at all times in order to receive data. Infrared remote controls (such as universal remote controls for televisions) are used to send commands to an infrared receiver connected to a microcontroller, which then interprets these signals and controls the robot's actions.

5.8.4 Radio Frequency (RF)

Commercially available **Remote Control (R/C) units** use small microcontrollers in the transmitter and receiver to send, receive, and interpret data sent via radio frequency (**RF**). The receiver box has a PCB (printed circuit board) which comprises the receiving unit and a small servo motor controller.

RF communication requires either a transmitter matched/paired with a receiver, or a transceiver (which can both send and receive data). **RF** does not require line of sight and can also offer a significant range (transmission distance). Standard radio frequency devices can allow for data transfer between devices as far away as several kilometers, and there is seemingly no limit to the range for more professional **RF** units.

Bluetooth is a form of **RF** and follows specific protocols for sending and receiving data. Normal **Bluetooth** range is often limited to about 10m, though it does have the advantage of allowing users to control their robot via Bluetooth-enabled devices such as cell phones, **PDAs**, and laptops (though custom programming may be required to create an interface). Just like **RF**, **Bluetooth** offers two-way communication.

WiFi is now an option for robots; being able to control a robot wirelessly via the internet presents some significant advantages (and some drawbacks) to wireless control. In order to set up a WiFi robot, you need a wireless router connected to the internet and a WiFi unit on the robot itself. For the robot, you can also use a device that is TCP/IP enabled with a wireless router.

Another wireless technology that was originally developed for human-to-human communication, the cell phone, is now being used to control robots. Since cellular frequencies are regulated, incorporating a cellular module on a robot usually requires added patience for programming as well as an understanding of the cellular network system and the regulation.

5.9 Autonomous Control

The next step is to use the microcontroller in your robot to its full potential and program it to react to input from its sensors. Autonomous control can come in various forms: pre-programmed with no feedback from the environment, limited sensor feedback, and finally, complex sensor feedback. True "autonomous control" involves a variety of sensors and code to allow the robot to determine by itself the best action to be taken in any given situation.

5.10 Sensors

i. **Switches, buttons, and contact sensors:** Used to detect physical contact between objects and are not just restricted to humans pushing buttons; bumpers on a robot can be equipped with momentary push buttons.

ii. **Pressure sensor:** Unlike a push button which offers one of two possible readings (ON or OFF), a pressure sensor produces an output proportional to the force that is being applied to it.

iii. **Ultrasonic Range Finder:** Uses acoustics to measure the time between when a signal is sent versus when its echo is received back. Ultrasonic range finders can measure a range of distances but are used specifically in air and are affected by the reflectivity of different materials.

iv. **Infrared light:** Can be used to measure distance. Some infrared sensors measure one specific distance while others provide an output proportional to the distance to an object.

v. **Laser:** Lasers are used when high accuracy, or long distances (or both) are required when measuring the range to an object. Scanning laser rangefinders use spinning lasers to get a two-dimensional scan of the distances to objects.

vi. **Encoders:** Optical encoders use mini infrared transmitter/receiver pairs and send signals when the infrared beam is broken by a specifically designed spinning disk (mounted to a rotating shaft). The number of times the beam is broken corresponds to the total angle travelled by a wheel. Knowing the radius of the wheel, you can determine the total distance travelled by that wheel. Two encoders give you a relative distance in two dimensions.

5.11 Robot Positioning

i. **Indoor Localization (room navigation):** An indoor localization system can use several beacons to triangulate the robot's position within a room, while others use a camera and landmarks.

ii. **GPS:** A GPS uses the signals from several satellites orbiting the planet to help determine its geographic coordinates. Regular GPS units can provide geographical positioning down to 5m of accuracy, while more advanced systems involving data processing and error correction thanks to the use of other GPS units or IMUs can be accurate down to several cm.

5.12 Tools

5.12.1 Mechanical Tools (Essential)

i. **Small screwdriver set:** These small screwdrivers are necessary when working with electronics. Don't force them too much, though – their size makes them more fragile.

ii. **Regular screwdriver set**: All workshops need a multi-tool or tool set that includes flat/Phillips and other screwdriver heads.

iii. **Needle-nose pliers: A** set of needle-nose pliers is incredibly useful when working with small components and parts and is a very inexpensive addition to your toolbox. These are different from regular pliers because they come to a point that can get into small areas.

5.13 Ultimate Tools

i. **Tabletop CNC mill:** A tabletop CNC machine allows you to precisely machine plastics, metals, and other materials and creates three-dimensional, intricate shapes.

ii. **Tabletop lathe:** A (manual) tabletop lathe allows you to create your own hubs, shafts, spacers, adapters, and wheels out of various materials. A CNC lathe tends to be overkill since most builders only need to change the diameter rather than create complex shapes.

iii. **Vacuum Forming Machine:** Vacuum forming machines are used to create complex plastic shells that are moulded to your exact specifications.

iv. **Metal Benders:** When making robotic frames or enclosures out of sheet metal or metal extrusions, using a metal bender is essential in order to obtain precise and repeatable bends.

v. **Other Specialized Tools:** At this stage, you will be very aware of your machining needs and will probably require more specialized tools, such as metal nibblers and welding.

5.14 Essential Tools

CAD Tools:

i. Google SketchUp is a free program that can be used to create your robot in 3D, to the proper scale, complete with texture. This can help you ensure that parts are not overlapping, check dimensions for EAGLE holes, and change the design before it is built.

ii. Autodesk 123D is another free 3D **CAD** (Computer Aided Design) software aimed at hobbyists. While it shares many of the same features as Google SketchUp, it has some interesting features such as solid-based part design, assemblies, parametrized transforms, and other functionalities that are usually seen in higher-end **CAD** programs.

5.15 Software Tools

i. **Programming software:** Your first programming software should correspond to whichever **microcontroller** you selected. If you chose an Arduino microcontroller, you should choose the Arduino Software; if you chose a Basic Stamp from Parallax, you should choose **PBasic**, and so forth. In order to use a variety of microcontrollers, you may want to learn a more fundamental programming language such as BASIC or C.

ii. **Schematics and PCBs:** There are many free programs available on the market, and **CadSoft's** is one of the more popular. It includes an extensive library of parts and helps you convert your schematic to a PCB.

Ultimate Tool:

i. **CAD SolidWorks** is the **CAD** program of choice for many when doing mechanical design, but it is certainly not the only one available. When working at this level (i.e., using programs worth several thousands of dollars), you should have a good idea of your needs in order to choose the right tool **(Unigraphics, Catia, ProE, etc.).**

ii. **CAM:** If you are using a **CNC** machine, you will need a proper **3D CAD** program such as **ProE**, **AutoCAD**, SolidWorks, or other similar programs. In order to convert your **CAD** model to usable code to send to the **CNC** machine, you need a **CAM** program. Often, you can purchase a **CAM** program specifically for the **CAD** software you selected, or find a third-party supplier.

5.16 Raw Materials (Essential)

i. **Thin sheet metal:** This material can be cut easily with scissors and can be bent and shaped as needed to form the frame or other components of your robot without necessarily having to do machining.

ii. **Cardboard:** The right cardboard (thick but can still be cut using hand tools) can easily be used to make a frame or prototype. Even basic glue can be used to hold cardboard together.

iii. **Thin plastic:** Polypropylene, PVC about 1/16" thick can be sawed to create a more rigid and longer-lasting frame for your robot.

iv. **Thin wood:** Wood is a great material to work with if you have the means. It can be screwed, glued, sanded, finished, and more.

v. **Polymorph:** Polymorph allows you to create plastic parts without the hassle of having to create custom moulds.

vi. **Sheet metal:** If you have thicker metal-cutting shears, sheet metal makes an excellent building material for a robot frame because of its durability, flexibility, and resistance to rust.

vii. **Plastic sheets:** Plastic sheets are fairly rigid and resist deformation. If you are cautious and slow when cutting or drilling most plastics, the results can look professional.

5.17 Assembling the Robot Components

i. Connect motors to motor controller.

ii. Connect batteries to a motor controller or a microcontroller. Most motor controllers have two screw terminals for the battery leads labelled B+ and B-. If your battery came with a connector and your controller uses screw terminals, you may be able to find a matching connector with pigtails (wires) which you can connect to the screw terminals. If not, you may need to find another way to connect the battery to the motor controller while still being able to unplug the battery and connect it to a charger.

It is possible that not all the electromechanical products you chose for your robot can operate at the same voltage and thus may require several batteries or voltage regulation circuits.

Usual voltage levels involved in common hobby robotics components:

i. DC gear motors – 3V to 24V

ii. Standard Servo motors – 4.8V to 6V

iii. Specialty Servo motors – 7.4V to 12V

iv. Stepper motors – 6V to 12V

v. Microcontrollers usually include voltage regulators – 3V to 12V

vi. Sensors – 3.3V, 5V, and 12V

vii. DC motor controllers – 3V to 48V

viii. Standard batteries – 3.7V, 4.8V, 6V, 7.4V, 9V, 11.1V, and 12V

Other connections:

i. Connect motor controllers to microcontroller.

ii. Connect sensors to a microcontroller.

iii. Connect communication device to microcontroller.

iv. Connect electrical components to frame.

5.18 Programming the Robot

There are many programming languages that can be used to program microcontrollers, the most common of which are:

i. **Assembly:** Just one step away from machine code and as such it is very tedious to use. Assembly should only be used when you need absolute instruction-level control of your code.

ii. **Basic:** One of the first widely used programming languages, it is still used by some microcontrollers (Basic Micro, BasicX, Parallax) for educational robots.

iii. **C/C++:** One of the most popular languages, C provides high-level functionality while keeping good low-level control. RobotC is a good example.

iv. **Java:** More modern than C and provides lots of safety features to the detriment of low-level control. Some manufacturers like Parallax make microcontrollers specifically for use with Java.

v. **.NET/C#:** Microsoft's proprietary language used to develop applications in Visual Studio. Examples include Netduino, FEZ Rhino, etc.

vi. **Processing Arduino:** A variant of C++ that includes some simplifications in order to make programming easier.

vii. **Python:** One of the most popular scripting languages. It is very simple to learn and can be used to put programs together very fast and efficiently.

5.19 Programming Skill Advice

i. **Create manageable chunks of functional code:** By creating segments of code specific to each product, you gradually build up a library. Develop a file system on your computer to easily look up the necessary code.

ii. **Document everything within the code using comments:** Documenting everything is necessary in almost all jobs, especially robotics. As you become more advanced, you may add comments to general sections of code, though as you start, you should add a comment to (almost) every line.

iii. **Save different versions of the code:** Do not always overwrite the same file. If you find one day that your 200+ lines of code do not compile, you won't be stuck going through it line by line; instead, you can revert to a previously saved (and functional) version and add/modify it as needed. Code does not take up much space on a hard drive, so you should not feel pressured to save only a few copies.

iv. **Raise the robot off the table or floor when debugging:** This prevents wheels/legs/tracks from accidentally launching it off the edge. Keep the power switch close by in case the robot tries to destroy itself. For example, if you send a servo motor a 400us signal when it only accepts a 500 (corresponding to 0 degrees) to 2500us (corresponding to 180 degrees) signal, the servo would try to move to a location it cannot physically go to (-9 degrees) and ultimately burn out.

v. **Turn off power if code misbehaves:** If code does something that does not seem to be working correctly after a few seconds, turn off the power. It is highly unlikely the problem will "fix itself," and in the meantime, you may be destroying part of the mechanics.

vi. **Use subroutines:** Subroutines may be a bit difficult to understand at first, but they greatly simplify your code. If a segment of code is repeated many times within the code, it is a good candidate to be replaced with a subroutine.

5.20 Programming the Robot

A good example of how program is used to move the robot can be illustrated thus:

i. motors(1,1) forward full speed

ii. motors(0.5,0.5) forward half speed

iii. motors(-1,-1) backward

iv. motors(0,0) / stop()

v. motors(1,-1) spin

Commands for pictures and movies:

programming command

```
pic = takePicture()
  show(pic)
savePicture(pic, "pic.jpg")
```

Robots have a series of different sensors to obtain information from the surroundings, such as:

i. **IR** sensors: locate various obstacles

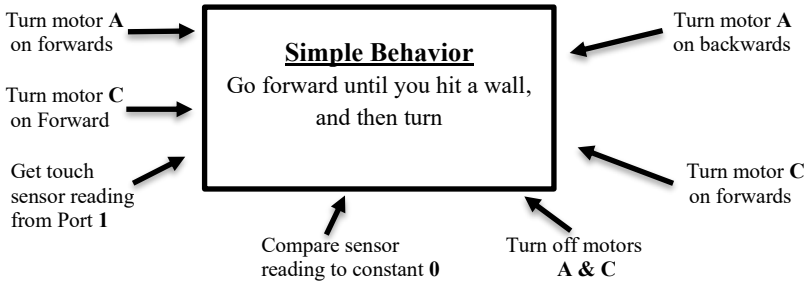
ii. Light sensors: measure light levels

5.21 Robot Programming Behaviours

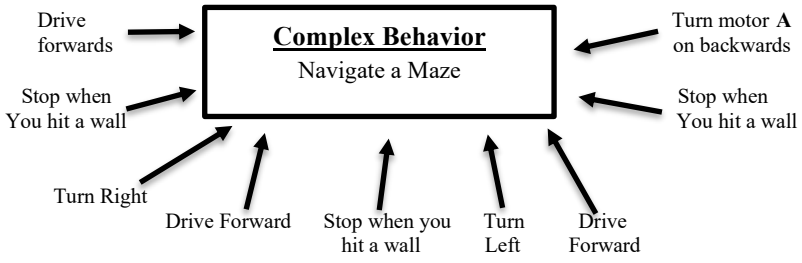
Robot programming behaviours describe the actions and decisions of your robot.

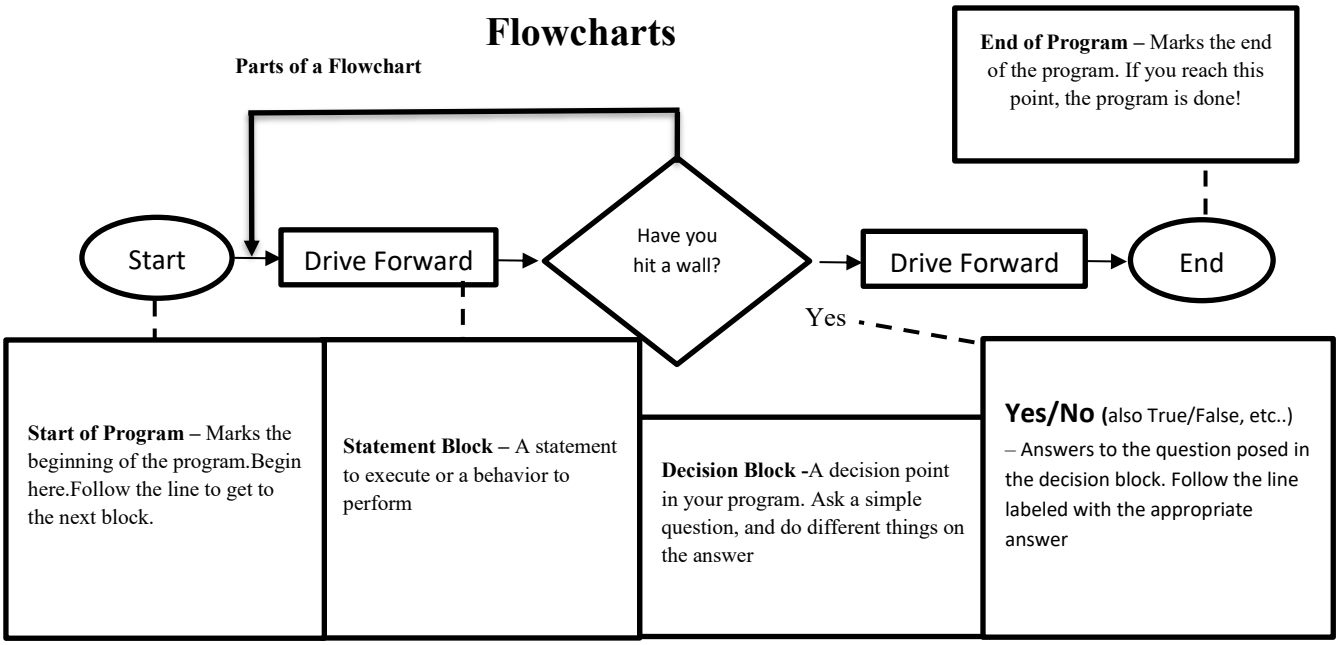
i. **Basic behaviours:** Actions the robot performs directly.
ii. **Simple behaviours:** Actions as we think about them.
iii. **Complex behaviours:** Describe more interesting functions of the robot.

Flowcharts organize the steps. Pseudocode describes the steps using the robot's motors and sensors.



Complex behaviour example:
Complex behaviours allow robots to describe the full scope of what the robot can do. This is always composed of smaller behaviours, so you can break them down.





Sample pseudocode:
Pseudo-code is halfway in between English and computer code. It describes what sensors and motors do and see with simple behaviors. Below is a sample of pseudocode:

i. Move forward until the touch sensor is pressed
ii. Stop all motors
iii. Reverse for 1 second
iv. Stop all motors
v. Turn right

5.22 Programming Language – ROBOTC

ROBOTC [3] is a text-based robot compiler. Commands to the robot are first written as text on the screen. They are then processed by the ROBOTC compiler into a machine language file that the robot can understand. Finally, they are loaded onto the robot, where they can be run via embedded systems programming. This is demonstrated in Fig. 2 below

Program code
(Comands to the robot written as text)

```
task main()
{

        motor[port3] =  127;
        wait1Msec(3000);

}
```

Fig 2: Sample ROBOTC Program – Motor Control
("motor[port3] = 127; " instructs the motor plugged into Motor Port 3 to turn on at full power.)

5.23 Code

Text written as part of a program is called code. You type code just like you type normal text. Keep in mind that capitalization is important to the computer. Replacing a lowercase letter with a capital letter (or vice versa) will cause the robot to become confused. See Fig 3:

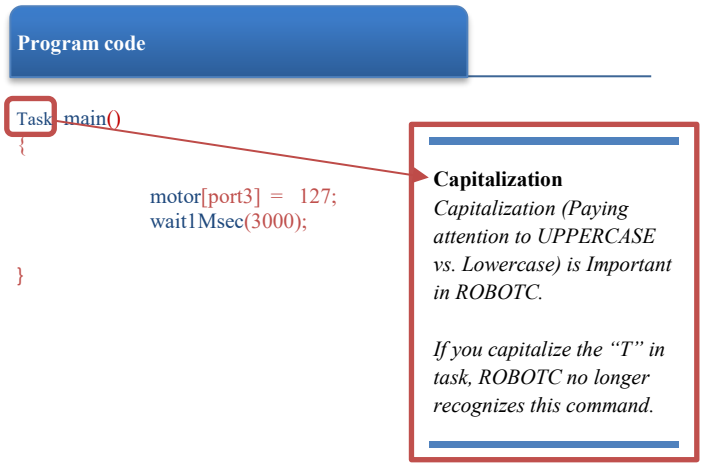


Fig. 3: Capitalization in **ROBOTC**

As you type your codes, **ROBOTC** will try to help you out by coloring the words it recognizes. If a word appears in a different color, it means **ROBOTC** recognizes it as an important word in the programming language.

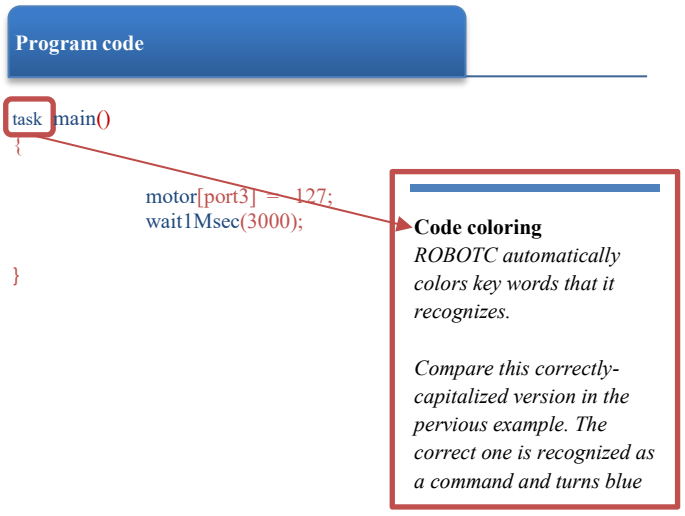
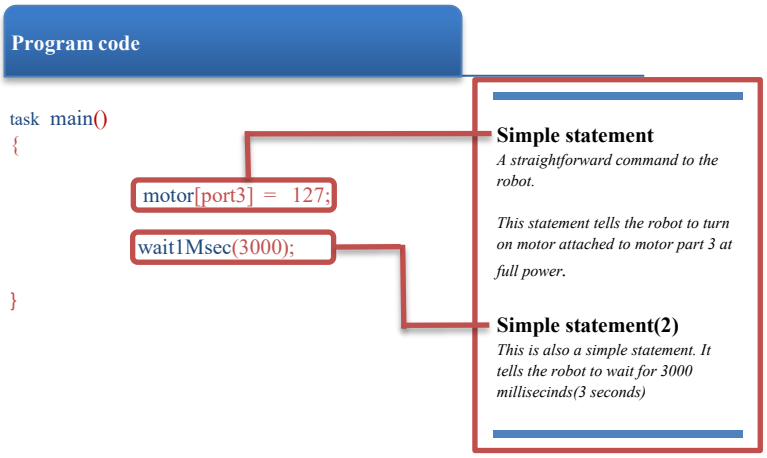


Fig. 4: Code Coloring in ROBOTC

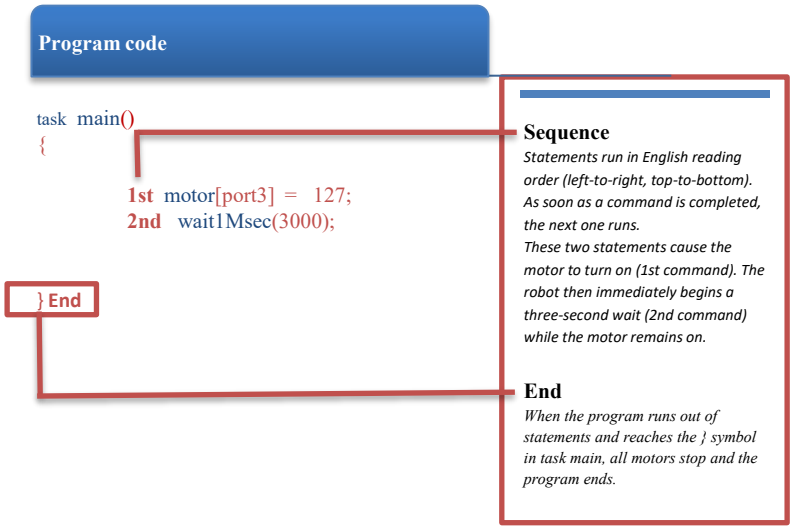
Statements:
Statements are instructions for the robot. The most basic kind of statement in ROBOTC simply gives a command to the robot. Example:



motor[port3] = 127; instructs the motor plugged into Motor Port 3 to turn on at full power.

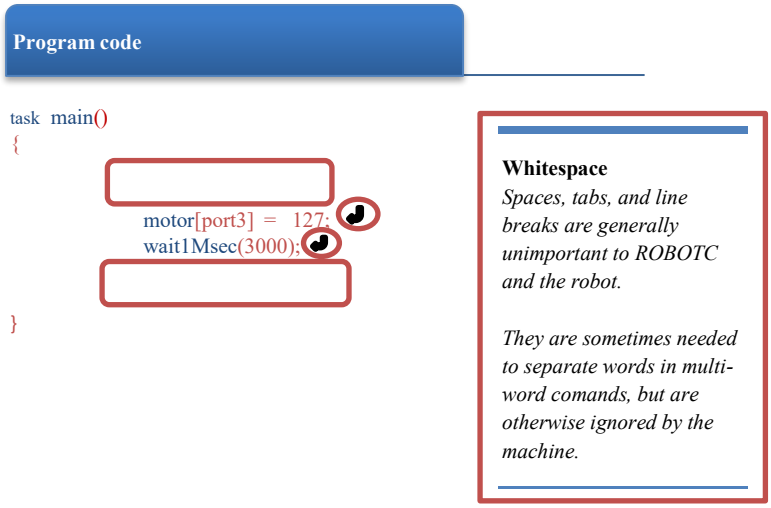
5.24 Running the Program

Statements are run in order as quickly as the robot is able to reach them. Running this program below on the robot turns the motor on, waits for 3000 milliseconds (3 seconds) with the motor still running, and then ends.



5.25 RobotC Rules

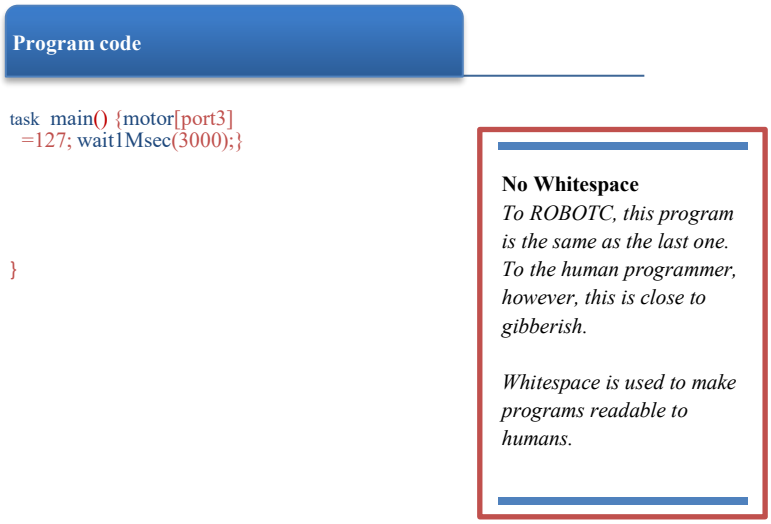
RobotC rules include the fact that white spaces, tabs, and line breaks are not important to the compiler and the robot but are sometimes needed for separation of words in multi-word commands. They are usually ignored by the robot. Thus, spaces and line breaks in ROBOTC are only used to separate words from each other in multi-word commands. Spaces, tabs, and lines do not affect the way a program is interpreted by the machine.



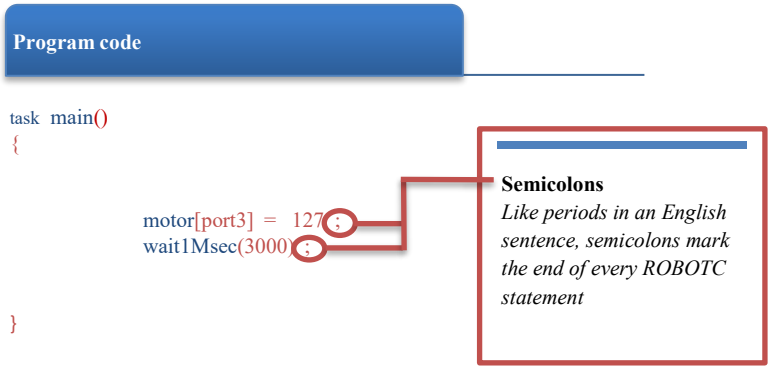
5.26 Programming Readability

Codes are written on separate lines to enable the programmer to interact and communicate with the robot. Using spaces, tabs, and lines helps human programmers read the code more easily.

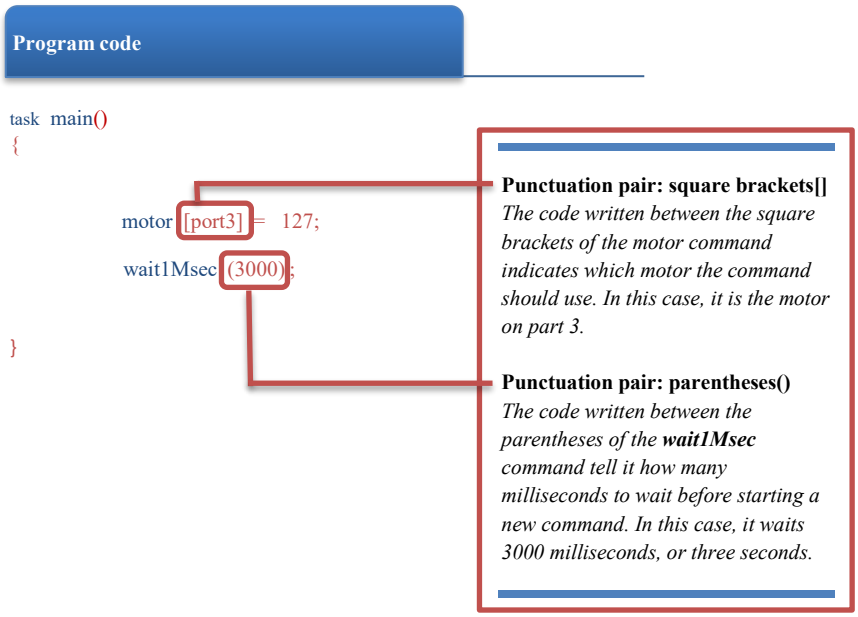
Making good use of spacing in your program is a very good habit for your own sake and for general readability. Observe the difference between Figs 7 and 8.



Punctuation:
Punctuation marks enable ROBOTC to know where one statement ends and the other begins. This is made possible by the semicolon (;) at the end of each line. Every statement ends with a semicolon. It is like the period at the end of a sentence.



Punctuation pairs:
Like parentheses and square brackets, punctuation pairs are used to mark off special areas of code. Every punctuation pair consists of an opening punctuation mark and a closing punctuation mark. The punctuation pair designates the area between them as having special meaning to the command that they are part of. See Fig 10



5.27 Motor Commands and Control Structures

Different commands make use of different kinds of paired punctuation. The motor command uses square brackets, and the wait1Msec command uses parentheses. This is just the way the commands are set up. You will have to remember to use the right punctuation with the right commands.

Control Structures:
Simple statements do the work in ROBOTC, but control structures do the thinking. Control structures (or control statements) are pieces of code that control the flow of the program's commands, rather than issue direct orders to the robot.

5.28 Comments

Comments are text that the program ignores. A comment can contain notes, messages, and symbols that may help a human, but would be meaningless to the robot. ROBOTC simply skips over them. Comments appear in green in ROBOTC.

6.0 Summary and Conclusion

Learning the skill to program robots will enable African industries to embrace robotics and encourage competition in terms of precision, quality, and price of manufactured goods coming out of the continent. Africa cannot develop without this caliber of knowledge workers because development is triggered by innovation in science and technology.

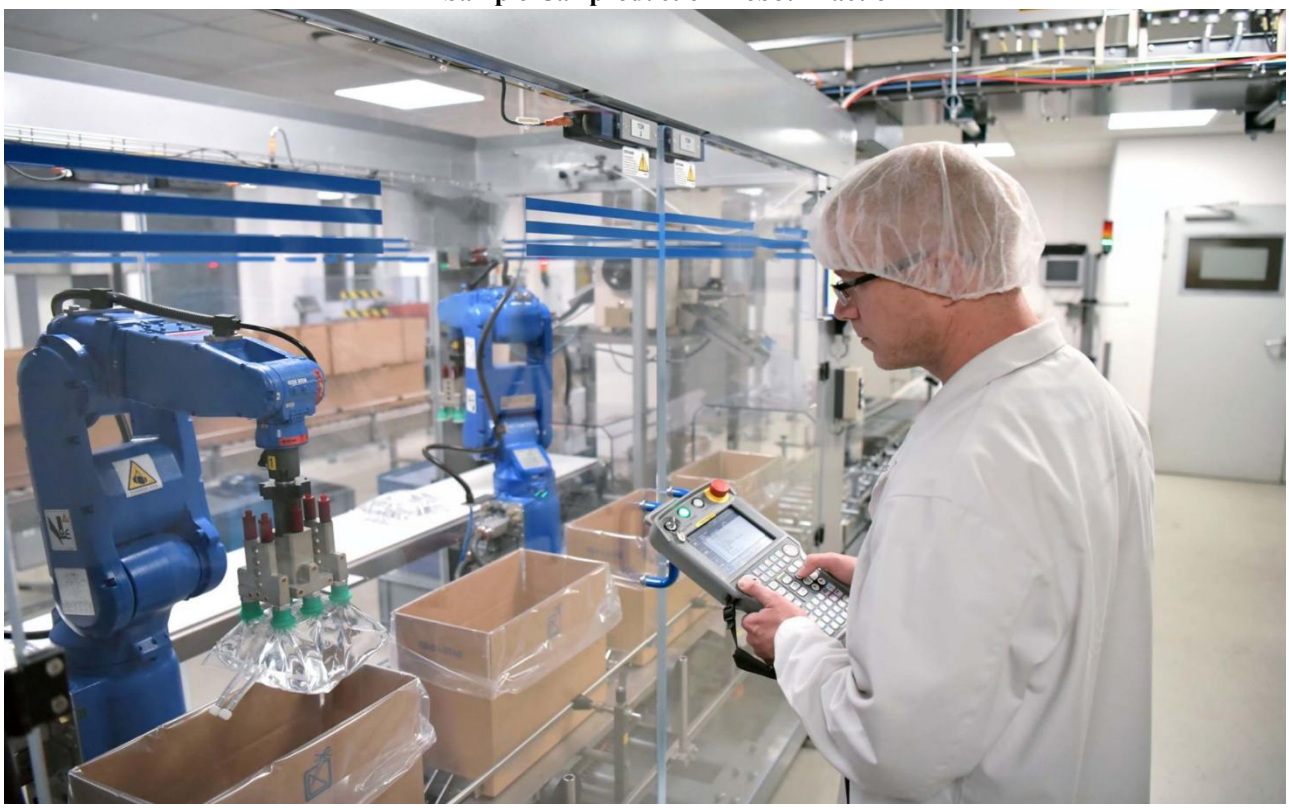
It is therefore recommended that African governments should endeavor to develop interest in the design, manufacture, and programming of intelligent robots for industry. This will create the right snowballing effect on our industries, particularly in the areas of market competition and penetration, product quality, pricing, and packaging. Can this paper generate interest in pragmatic automation of African industries? That is my hope.

References

[1]. Robot Institute of America http://www.ri.cmu.edu/
[2]. https://www.robots.com/articles
[3]. https://www.robotc.net/
[4]. https://www.python.org/doc
[5]. https://docs.python.org/3.3/tutorial/introduction.html#using-python-as-a-calculator
[6]. http://www.robotshop.com/blog/en/how-to-make-a-robot-lesson-3-actuators-2-370

7.0 Appendices of some Industrial Robots in Action

Sample Car production Robot in action



Robots in pharmaceutical production

China boosts hybrid crop development speed by 400% with world-first robotic breeding plant