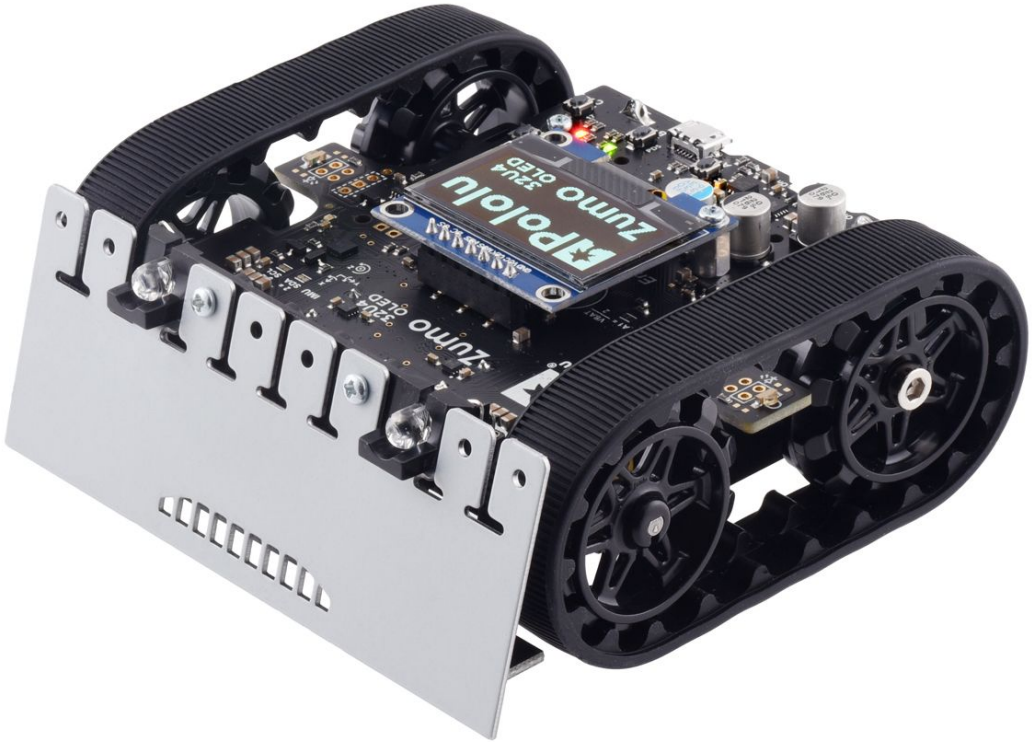
Autonomous Systems Lab - Embedded

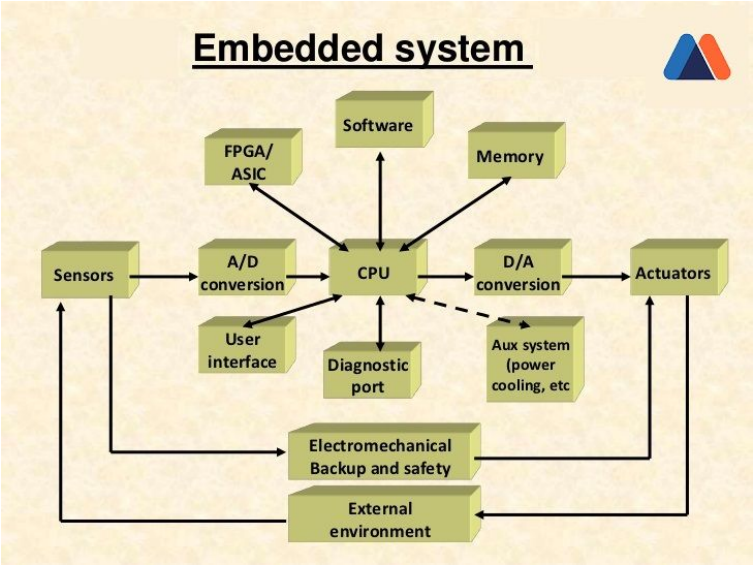


What is an Embedded System?

An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations.

The basic structure of an embedded system includes the following components:

- Sensor - Measures and converts the physical quantity
- A-D Converter - Analog to Digital Converter
- Processor & ASICs - Processes the data
- D-A Converter - Digital to Analog Converter
- Actuator - Acts on the External environment



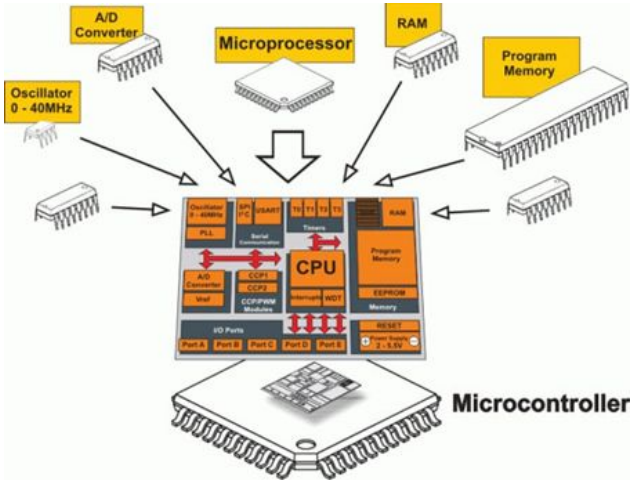
By: https://www.heavy.ai/technical-glossary/embedded-systems , https://swathiprabhala.blogspot.com/2019/02/embedded-systems.html

Micro Controller Unit (MCU)

A microcontroller is a single Integrated Circuit (IC) that is typically used for a specific application and designed to implement certain tasks. Products and devices that must be automatically controlled in certain situations, like appliances, power tools, automobile engine control systems, and computers are great examples, but microcontrollers reach much further than just these applications. a microcontroller gathers input, processes this information, and outputs a certain action based on the information gathered.

A microcontroller can be seen as a small computer with all the essential components inside of it

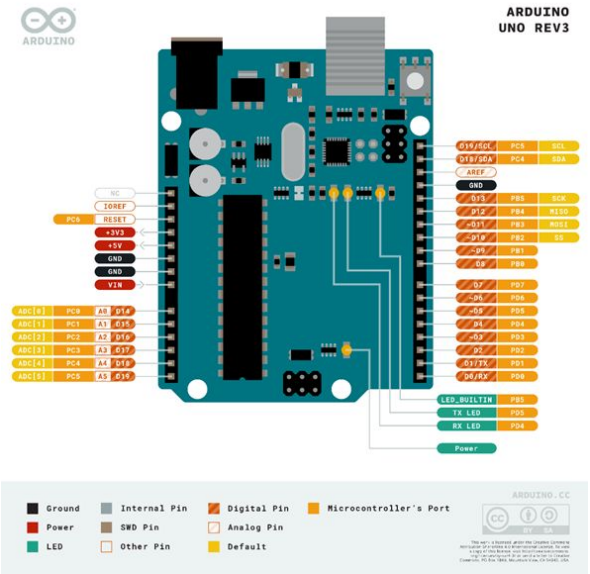
- Central Processing Unit (CPU)
- Random-Access Memory (RAM)
- Flash Memory
- Serial Bus Interface
- Input/Output Ports (I/O Ports)
- Electrical Erasable Programmable Read-Only Memory (EEPROM)



By: https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller

Arduino Development Board

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs, amount of light on a sensor, a press on a button, or the arrival of a Twitter message and turn this into an output.



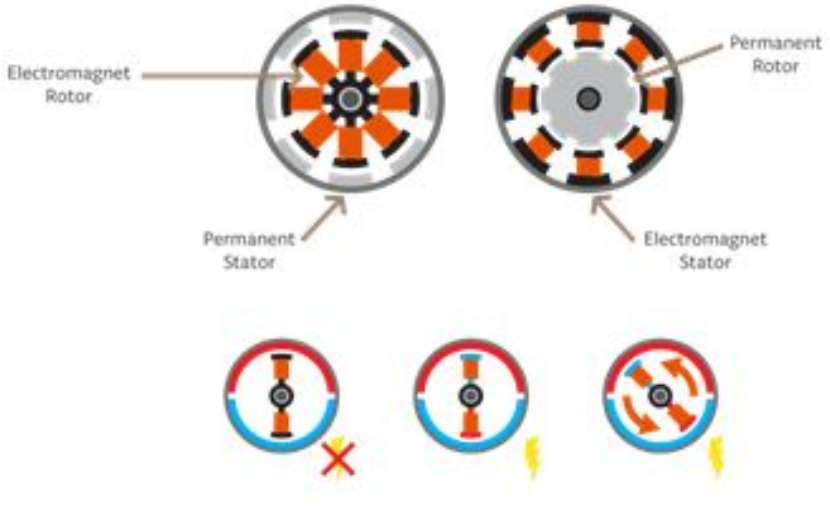
By: https://www.arduino.cc/

Arduino Uno specifications:

Microcontroller	ATmega328P	
USB connector	USB-B	
Pins	**Built-in LED Pin**	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	**UART**	Yes
	I2C	Yes
	SPI	Yes
Power	**I/O Voltage**	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Power Supply Connector	Barrel Plug
Clock speed	**Main Processor**	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	**ATmega328P**	2KB SRAM, 32KB FLASH, 1KB EEPROM

DC Motor

A DC (Direct Current) motor is a type of motor that will cause the motor shaft to rotate around its longitudinal axis when applying an electric current between its terminal pins. Thus, the DC motor is a type of actuator that transforms electrical current into rotational motion. The DC motor spins when we apply DC voltage through its two terminal pins. We can vary the speed of the motor by changing the voltage level. Motors can run in both directions just by reversing the direction of the current.

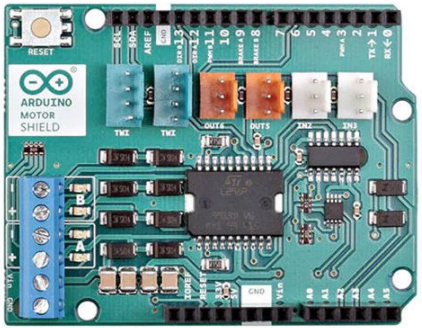


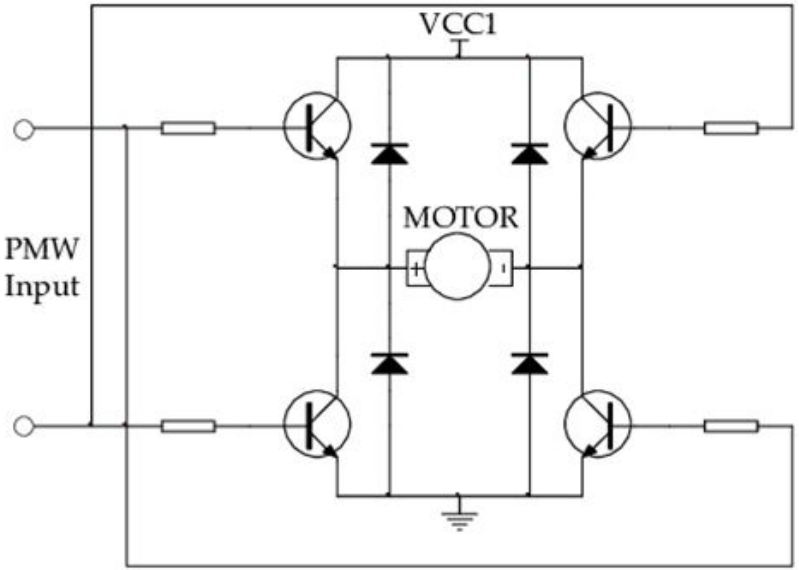
By: https://www.arduino.cc/

Motor Driver H-Bridge

Most electric motors need a higher voltage than can be provided by a microcontroller. To control a DC motor from a microcontroller, you'll need to use a driver. This can be a transistor, a relay or an H-Bridge. The H-Bridge is an electronic circuit that contains four transistors arranged so that the current can be driven to control the direction of the spin and the angular speed.

With the Arduino IDE, if you want to just turn a DC motor on and off, you can use a digitalWrite command with HIGH for ON and LOW for OFF. If you instead want to modify the speed of the motor, you can use the analogWrite command and one of the PWM pins.



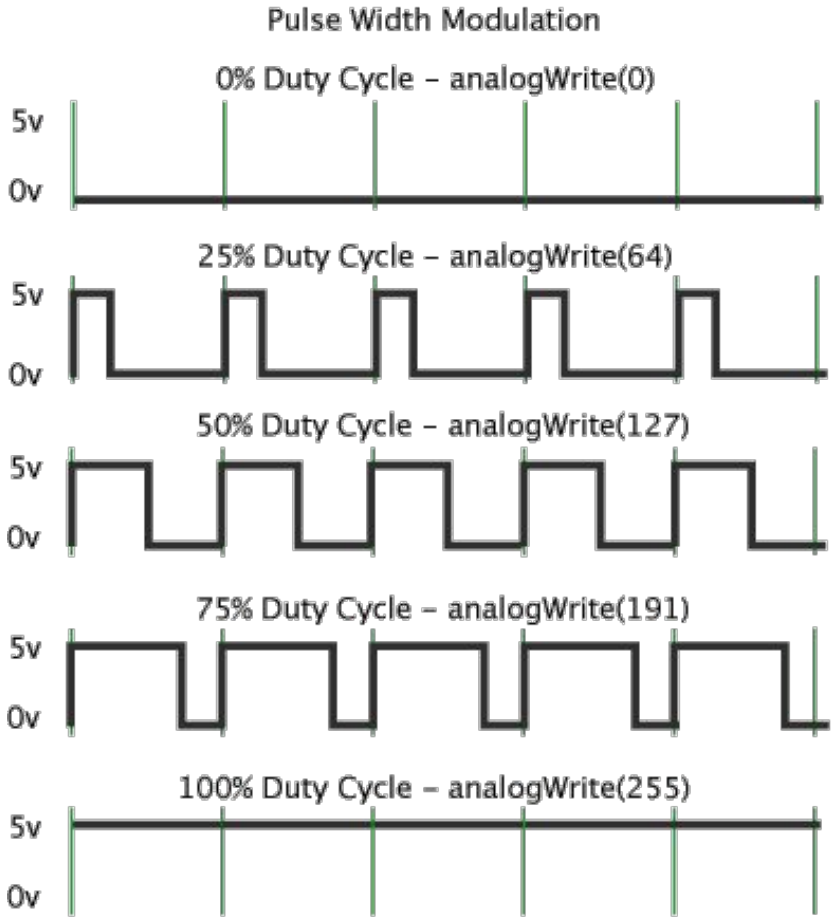


By: https://www.arduino.cc/

PWM

Pulse Width Modulation, or PWM, is a digital modulation technique commonly used to control the power supplied to electrical devices, like motors. The modulation technique consists of changing the width of a periodic signal's pulse. The width of the pulse is referred to as the duty-cycle and goes from 0% (minimum width) to 100% (maximum width).

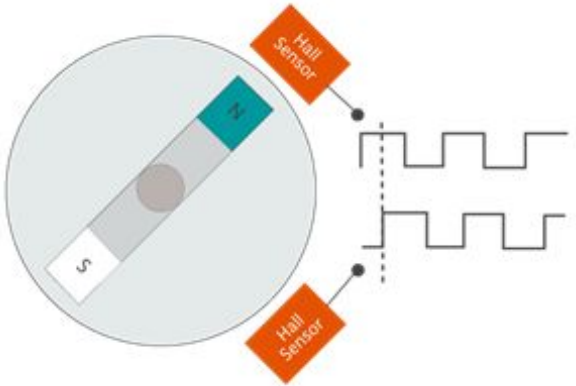
Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time)



By: https://www.arduino.cc/

Encoder

Magnetic encoders are sensors that can report information about the rotation speed and the spinning direction of the motor when mounted on a motor. The magnetic encoders are composed of a module with two Hall-effect sensors and magnetic discs. As the motor turns, the disc rotates past the sensors. Each time a magnetic pole passes a sensor, the encoder outputs a digital pulse, also called a “tick”. By counting those ticks, the speed of the motor can be determined.

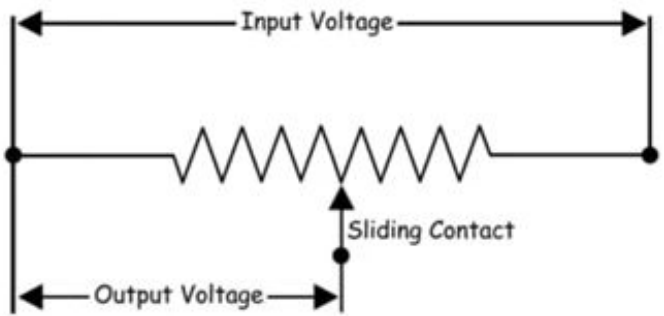


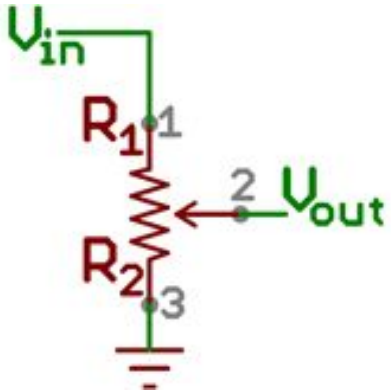
By: https://www.arduino.cc/

Potentiometer / Analog to Digital Converter (ADC)

A potentiometer (also known as a pot or potmeter) is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider.

An Analog to Digital Converter (ADC) converts an analog voltage on a pin to a digital number. One of the most common techniques uses the analog voltage to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete.



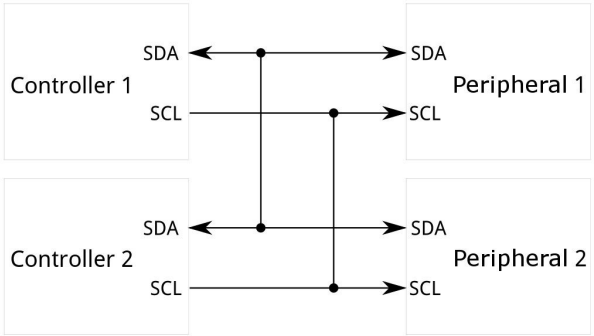


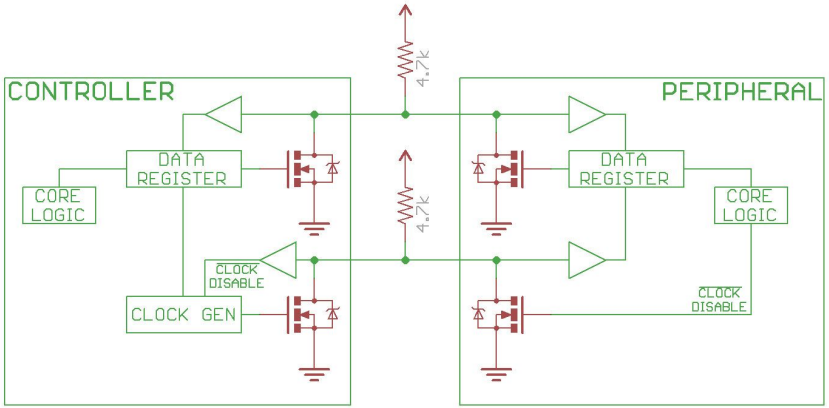
$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2}$$

By: https://www.arduino.cc/ , https://learn.sparkfun.com/tutorials

I2C communication protocol

The Inter-Integrated Circuit (I2C) Protocol is a protocol intended to allow multiple "peripheral" digital integrated circuits ("chips") to communicate with one or more "controller" chips. It is only intended for short distance communications within a single device. Like Asynchronous Serial Interfaces (such as RS-232 or UARTs), it only requires two signal wires to exchange information. It can support multiple peripheral devices. Each I2C bus consists of two signals: SDA and SCL. SDA (Serial Data) is the data signal and SCL (Serial Clock) is the clock signal. The clock signal is always generated by the current bus controller. The I2C bus drivers are "open drain", meaning that they can pull the corresponding signal line low, but cannot drive it high. Thus, there can be no bus contention where one device is trying to drive the line high while another tries to pull it low.

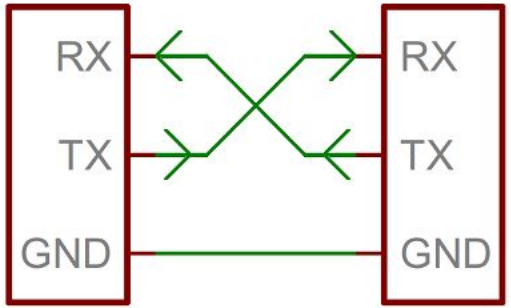




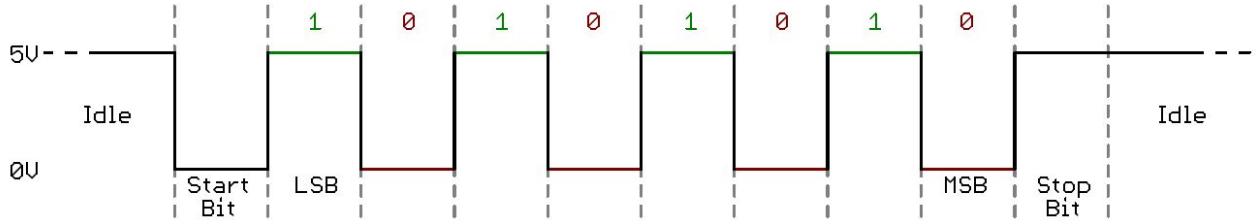
By: https://learn.sparkfun.com/tutorials

Serial communication protocol

A serial bus consists of just two wires - one for sending data and another for receiving. As such, serial devices should have two serial pins: the receiver, RX, and the transmitter, TX. When microcontrollers and other low-level ICs communicate serially they usually do so at a TTL (transistor-transistor logic) level. TTL serial signals exist between a microcontroller's voltage supply range - usually 0V to 3.3V or 5V.



The asynchronous serial protocol has a number of built-in rules - mechanisms that help ensure robust and error-free data transfers. These mechanisms, which we get for eschewing the external clock signal, are: Data bits, Synchronization bits, Parity bits, and Baud rate.

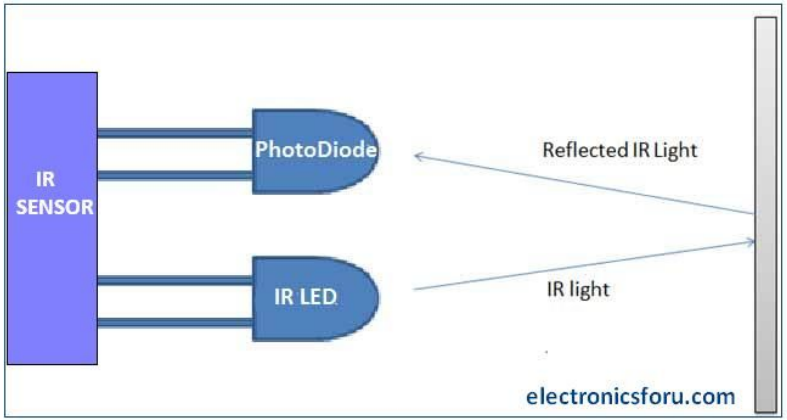


By: https://learn.sparkfun.com/tutorials

IR sensor

An IR sensor is an electronic device that detects IR radiation falling on it. An IR sensor consists of two parts, the emitter circuit and the receiver circuit. This is collectively known as a photo-coupler or an optocoupler. The emitter is an IR LED and the detector is an IR photodiode. The IR photodiode is sensitive to the IR light emitted by an IR LED. The photodiode`s resistance and output voltage change in proportion to the IR light received. This is the underlying working principle of the IR sensor. The type of incidence can be direct incidence or indirect incidence. In direct incidence, the IR LED is placed in front of a photodiode with no obstacle in between. In indirect incidence, both the diodes are placed side by side with an opaque object in front of the sensor. The light from the IR LED hits the opaque surface and reflects back to the photodiode.

Proximity sensors employ reflective indirect incidence principle. The photodiode receives the radiation emitted by the IR LED once reflected back by the object. Closer the object, higher will be the intensity of the incident radiation on the photodiode. This intensity is converted to voltage to determine the distance.

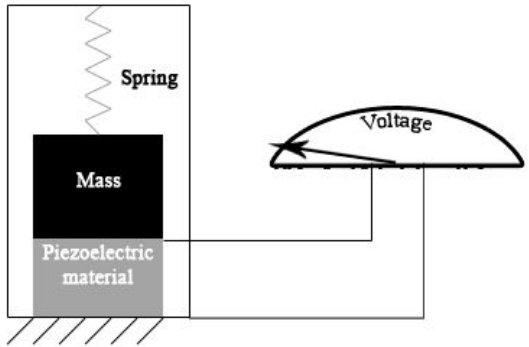


By: https://www.electronicsforu.com/technology-trends/learn-electronics/ir-led-infrared-sensor-basics

Accelerometer

Accelerometers are devices that measure acceleration, which is the rate of change of the velocity of an object. They measure in meters per second squared (m/s2) or in G-forces (g). A single G-force for us here on planet Earth is equivalent to 9.8 m/s2, but this does vary slightly with elevation (and will be a different value on different planets due to variations in gravitational pull). Accelerometers are useful for sensing vibrations in systems or for orientation applications. Accelerometers are electromechanical devices that sense either static or dynamic forces of acceleration. Static forces include gravity, while dynamic forces can include vibrations and movement

Accelerometers can measure acceleration on one, two, or three axes. 3-axis units are becoming more common as the cost of development for them decreases. Generally, accelerometers contain capacitive plates internally. Some of these are fixed, while others are attached to minuscule springs that move internally as acceleration forces act upon the sensor. As these plates move in relation to each other, the capacitance between them changes. From these changes in capacitance, the acceleration can be determined. Other accelerometers can be centered around piezoelectric materials. These tiny crystal structures output electrical charge when placed under mechanical stress (e.g. acceleration).

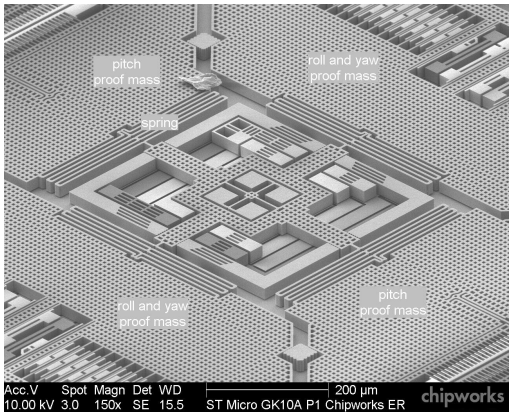


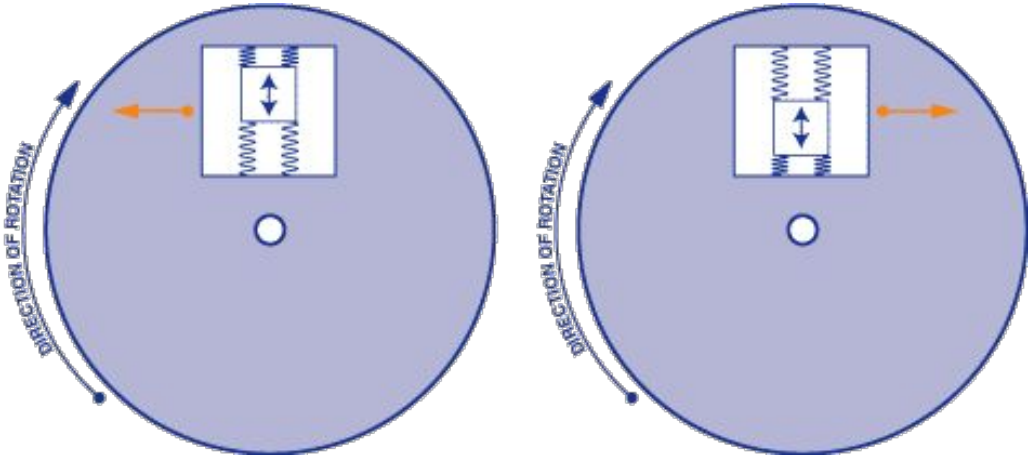
By: https://learn.sparkfun.com/tutorials/accelerometer-basics/all

Gyroscope

Gyroscopes, or gyros, are devices that measure or maintain rotational motion. MEMS (microelectromechanical system) gyros are small, inexpensive sensors that measure angular velocity. The units of angular velocity are measured in degrees per second (°/s) or revolutions per second (RPS). Angular velocity is simply a measurement of speed of rotation

The gyroscope sensor within the MEMS is tiny (between 1 to 100 micrometers, the size of a human hair). When the gyro is rotated, a small resonating mass is shifted as the angular velocity changes. This movement is converted into very low-current electrical signals that can be amplified and read by a host microcontroller.



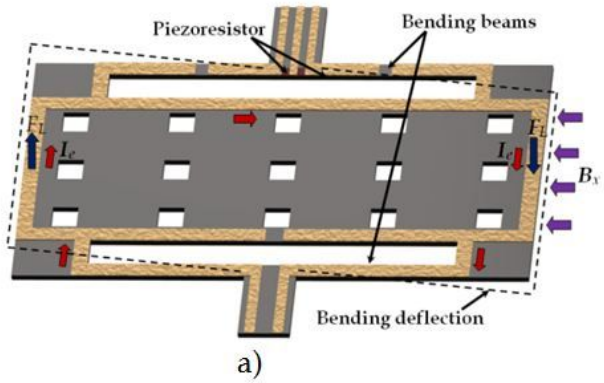


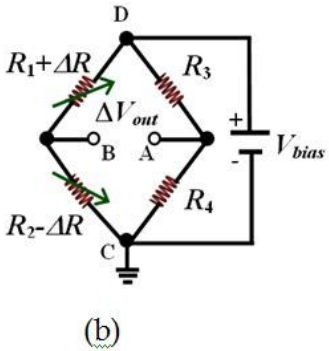
By: https://learn.sparkfun.com/tutorials/gyroscope/all , https://www.memsjournal.com/2011/01/motion-sensing-in-the-iphone-4-mems-gyroscope.html

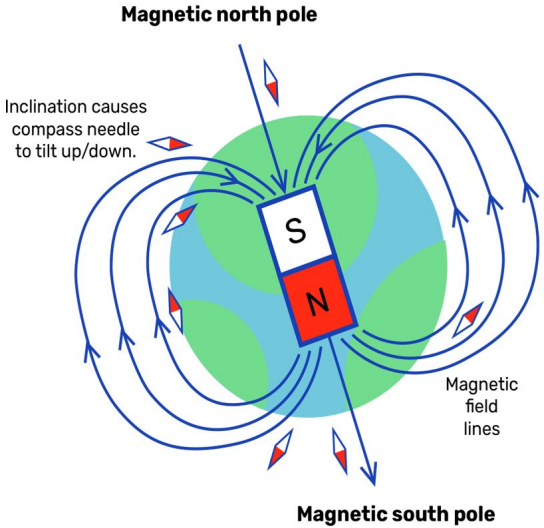
Magnetometer - Compass

A magnetometer is a device used to measure the magnetic field, particularly with respect to its magnetic strength and orientation. A popular example of a magnetometer would be the compass, which is used to measure the direction of an ambient magnetic field. Other magnetometers measure magnetic dipole moments; a magnetic dipole is the limit of either a closed loop of electric current or a pair of poles.

Resonant magnetic field microsensors based on MEMS employ resonant structures for monitoring magnetic fields through Lorentz force and use capacitive, piezoresistive or optical sensing techniques.







By: https://www.youngwonks.com/blog/What-is-a-Magnetometer-and-How-Does-It-Work , https://www.intechopen.com/chapters/15851

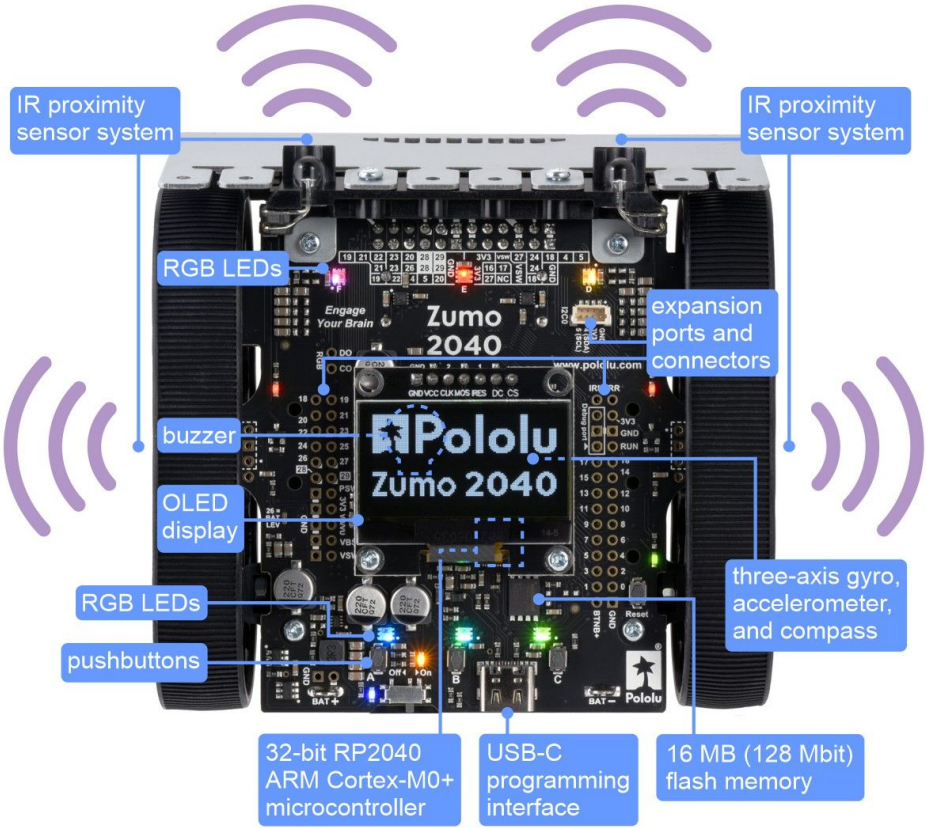
Zumo 2040

Zumo robot include a built-in Arduino-compatible RP2040 microcontroller, encoders for closed-loop motor control, and proximity sensors for obstacle detection. The robot is compact enough to qualify as a mini sumo robot, but its high-performance motors and integrated sensors make it versatile enough to serve as a general-purpose small robot.
Documentation

Links:

https://www.pololu.com/product/5012

https://github.com/pololu/zumo-2040-robot

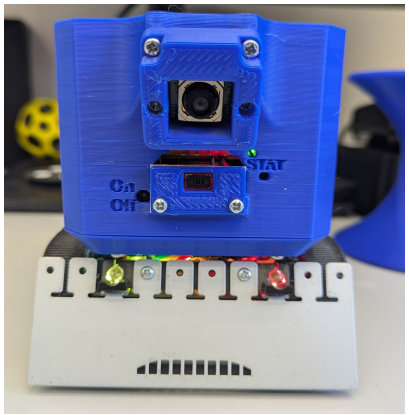


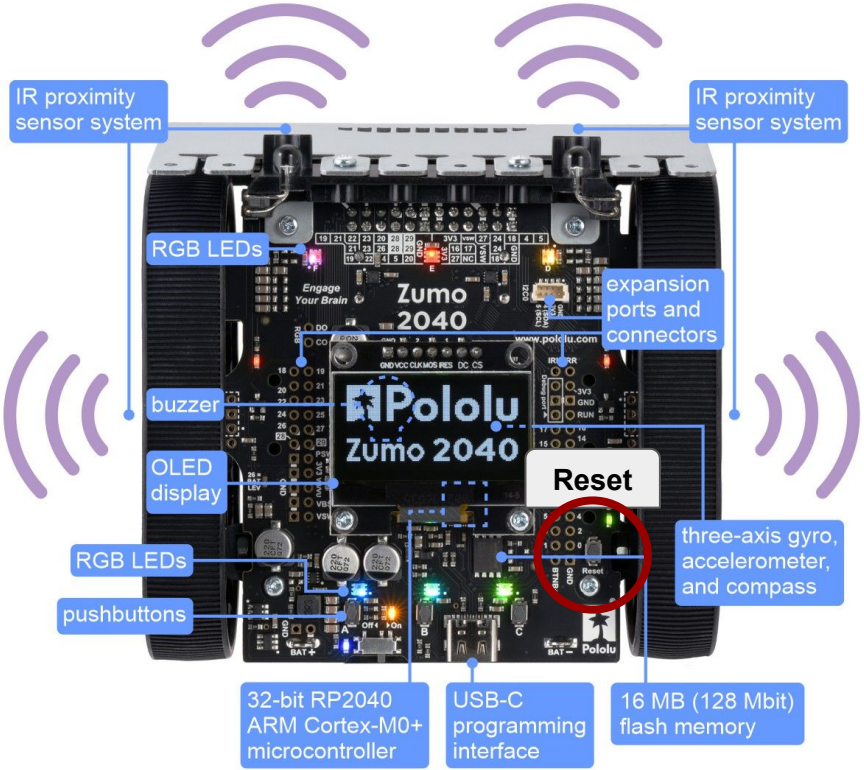
By: https://www.pololu.com/

Practice 0 - Turn On the ZumoPi V02 Platform

- Turn on the platform by pressing the On/Off button
- Press the Reset Button on the zumoPiV 02 (required after boot)
- Turn on the power slide on the zumo Platform

The ZumoPi V02 has a built in auto shutdown after 3 hours.





By: https://www.pololu.com/product/5012

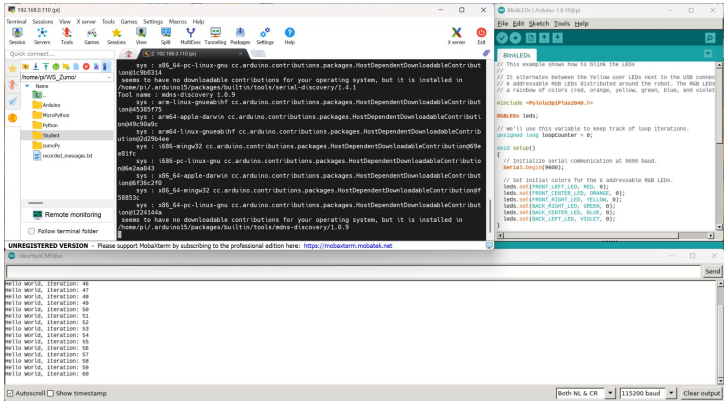
Practice 1 - Arduino code upload & Reminder

- Open mobaXterm and connect to the platform

  pi@192.168.0.1xx Password : zumo

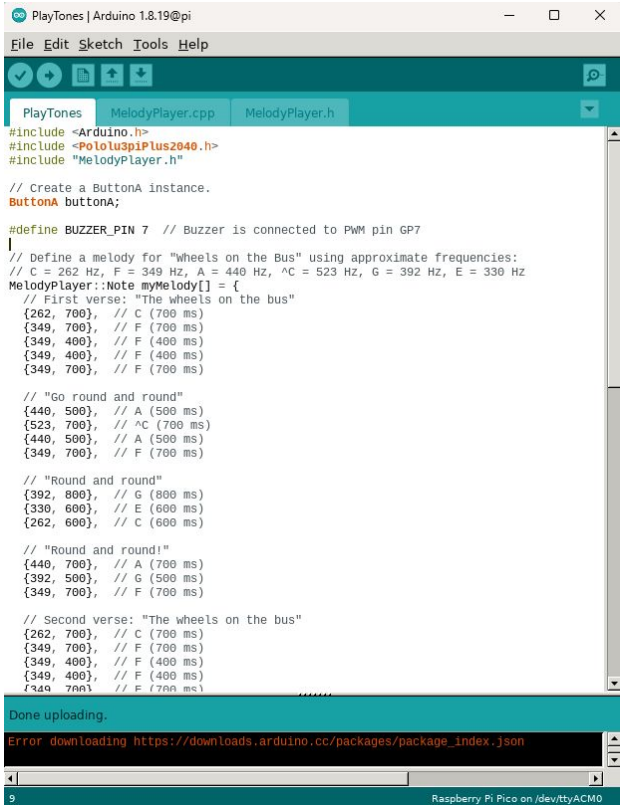
  - Run **arduino** command in the terminal
  - Open Led blink example File→Open
  - ~/WS_Zumo/Student/<Folder>/BlinkLEDs/BlinkLEDs.ino
  - Upload the the code and notice the Leds
- Upload code using Arduino CLI - (Command Line Interface)
  - Navigate to the code using a terminal
  - cd WS_Zumo/Student …
  - Compile and upload the code using aliased commands:

    (Close Serial Monitor between Methods)
    - zumoCompile BlinkLEDs.ino
    - zumoUpload BlinkLeds.ino



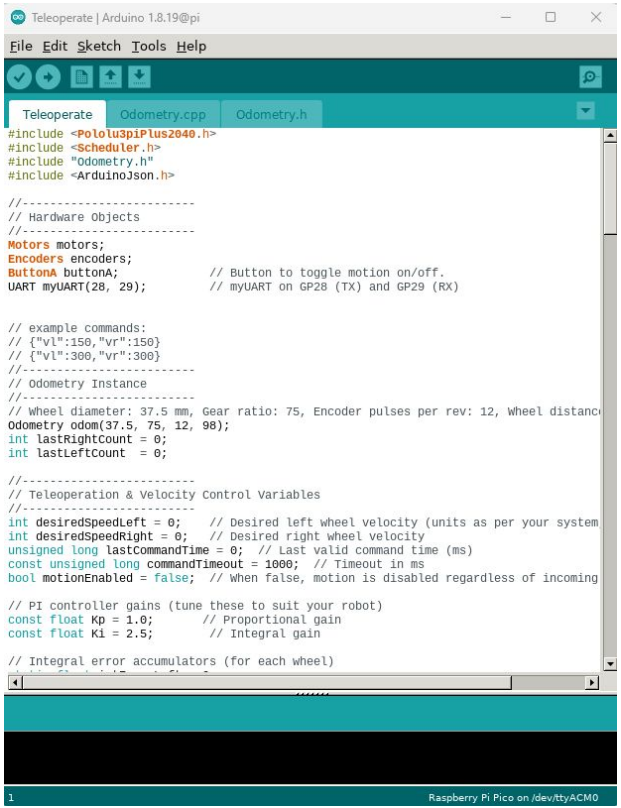
Exercise 1 - Buzzer

- Open the buzzer example for the zumo platform
  - ~/WS_Zumo/Arduino/Examples/PlayTones/PlayTones.ino
- Run the code Press A and enjoy the music, to stop the melody Press A again to stop the melody.
- Save the code under Student folder
  - ~/WS_Zumo/Student/<Folder>/…
- Modify the code to play a melody once, and a different melody, You can find one by googling “Popular songs in letter notes”
- Modify the code to play each time a button is pressed on the zumo robot (Button C).



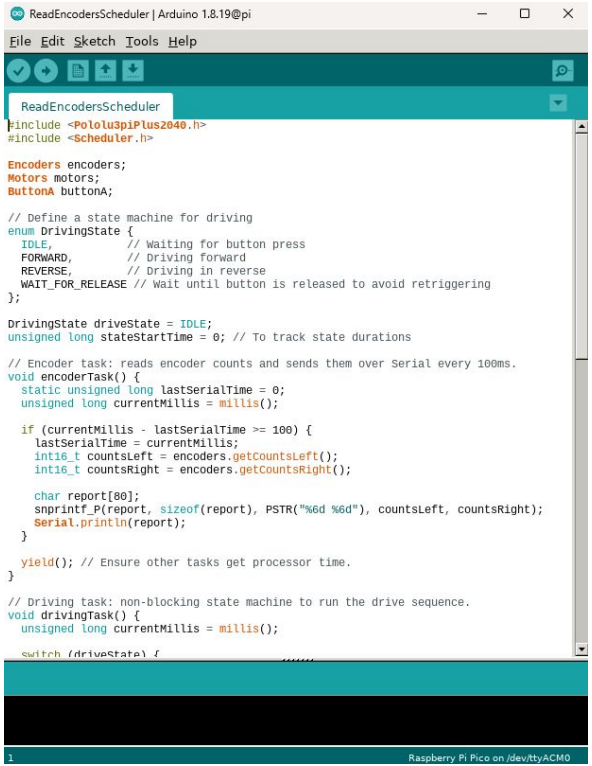
Practice 2 - Zumo TeleOperate - Review / Reminder

- Open TeleOperate example:
  - ~WS_Zumo/Arduino/Examples/TeleOperate/TeleOperate.ino
- Review the code
- Save a copy under your student directory
- Compile and run the code
- Run JoystickCamera.py directly or through and an aliased command **zumoJoystick**
- Using the browser navigate to the ip address of your platform using port 8500 i.e 192.168.0.110:8500
- Press Button A on the Zumo platform and switch on the power switch on the Zumo
- Move the platform and monitor the values
- **Stop the python script before proceeding**



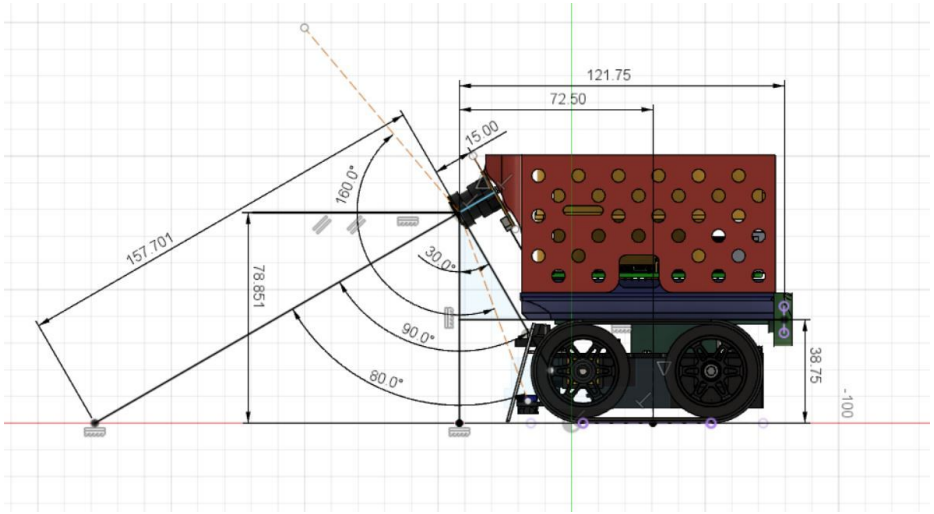
Practice 3 - Encoders example

- Open Zumo Encoder example:
  - ~WS_Zumo/Arduino/Examples/ReadEncoders/ReadEncoders.ino
  - Review the code, open Serial Monitor and run the example.
  - Modify the code to move only forward. Read the encoders and compare the measured distance against the encoder readings.
- Open Zumo Encoder Scheduler example:
  - ReadEncodersScheduler.ino
  - Review the code, and run the example. Understand the workflow of the Scheduler. And the resulting prints in Serial Monitor.



Exercise 2 - Encoders vs distance conversion

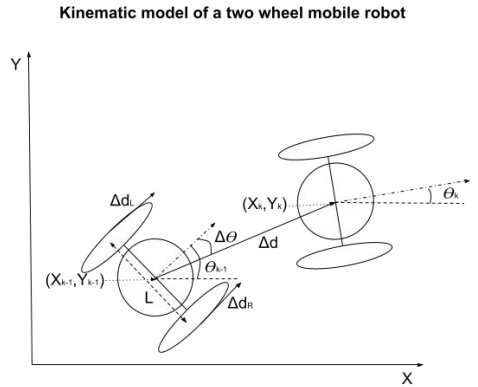
- Using the generated example evaluate the following parameters:
  - Encoder pulses to distance
  - Assuming the encoder has 12 pulses per revolution and the wheels diameter is 38.75 mm what is the gear ratio of the motor - https://www.pololu.com/docs/0J63/3.4
  - Hint: Distance = 2πr*NumPulses/(GearRatio*12)



Practice 4 - Odometry Example

- Open Zumo Odometry example:
  - ~WS_Zumo/Arduino/Examples/Odometry/Odometry.ino
  - Review the code, and run the example. Verify the distances measured, angle turned. Adjust gear ratio if needed.

Odometry is the process of estimating a rover's position and orientation by measuring wheel rotations with encoders and applying known dimensions (wheel diameter and distance between wheels) to calculate its movement. For additional read consider: PathControl



Exercise 3 - Open loop square motion

- Square motion using arduino
  - Open one of the motion examples reviewed earlier.
  - Modify the code to implement a square motion each time ButtonA is pressed
- Square motion using python with TeleOperate
  - Upload the TeleOperate.ino example to the platform
  - Create a python script to generate a square motion using the supplied example (next slide)

Notes:

- The Vl and Vr commands are in mm/s. Notice how teleoperate implements velocity control loop.
- There is a communication timeout implemented in case communication break up. So for longer motion periods consider sending the command in a for loop.
- Don`t forget to enable motion by pressing button A

Exercise 3 - Open loop square motion - Example code

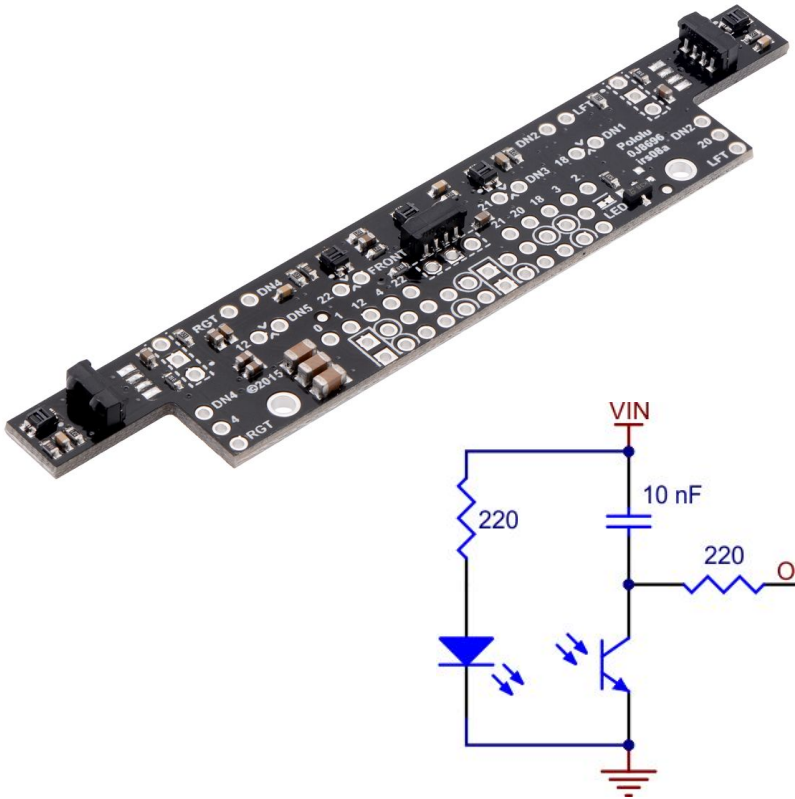
```
import serial
import time

# Open serial port
ser = serial.Serial('/dev/ttyAMA10', 115200, timeout=1)
time.sleep(1)  # Allow some time for the connection to initialize
# Move forward (both wheels at 150)
ser.write(b'{"vl":150,"vr":150}\n')
time.sleep(0.5)
# Rotate in place (left wheel forward, right wheel backward)
ser.write(b'{"vl":150,"vr":-150}\n')
time.sleep(0.5)
# Stop
ser.write(b'{"vl":0,"vr":0}\n')
ser.close()
```

Front Sensor Array

The Zumo 32U4 Front Sensor Array is a separate board that attaches to the main board. The board features five line sensors and three proximity sensors, though by default, you can only have six of these eight sensors connected to the Zumo's microcontroller at any given time.

The five line sensors face downward and can help the Zumo distinguish between light and dark surfaces.

The three proximity sensors face in different directions away from the Zumo and can help detect nearby objects.



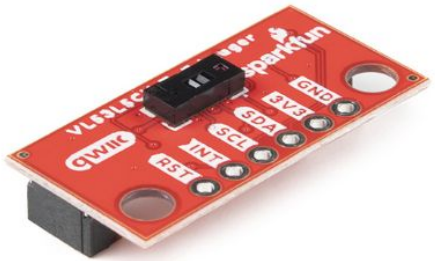
By: **https://www.pololu.com/product/3122**, **https://www.pololu.com/product/2459** , https://www.pololu.com/file/0J683/GP2S60_DS.pdf https://www.pololu.com/file/0J615/tssp77038.pdf , https://www.pololu.com/docs/0J63/3.6

Practice 5 - Front Sensor Array

- Copy example LineSensorTest.ino to your student folder.
  - ~WS_Zumo/Arduino/Examples/LineSensorTest/LineSensorTest.ino
  - Upload code and monitor the signals using Serial Monitor
  - Identify the values corresponding to the sensors

- Copy example LineFollowerSimple.ino to your student folder.
  - ~WS_Zumo/Arduino/Examples/LineFollowerSimple/LineFollowerSimple.ino
  - Upload code and monitor the signals using Serial Monitor
  - Modify the controller for best performance.

Practice 6 - Distance Sensor VL53L5CX



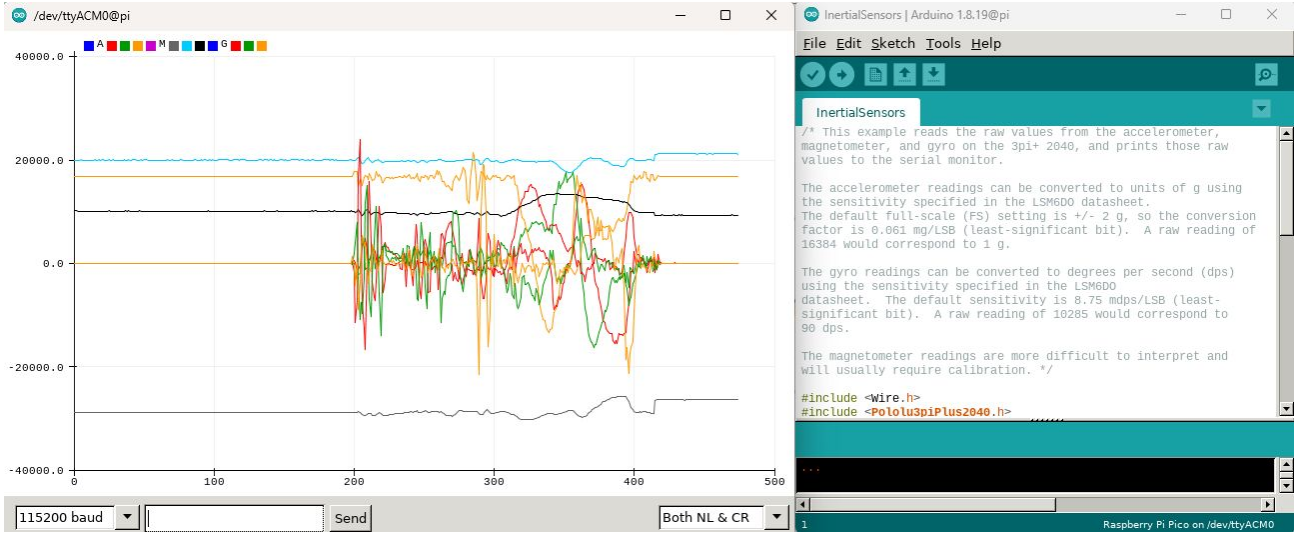
The SparkFun Qwiic Mini ToF Imager is a state of the art, 64 pixel Time-of-Flight (ToF) four meter ranging sensor built around the VL53L5CX from STMicroelectronics; A state of the art, Time-of-Flight (ToF), multi zone ranging sensor

- Copy example VL53L5CX_DistanceArray.ino to your student folder.
  - ~WS_Zumo/Arduino/Examples/VL53L5CX_DistanceArray/VL53L5CX_DistanceArray.ino
  - Upload code and monitor the signals using Serial Monitor
  - Change resolution to 8x8 notice the difference.
- Copy example TrackObject.ino to your student folder.
  - ~WS_Zumo/Arduino/Examples/TrackObject/TrackObject.ino
  - Review the code and run the example.
  - Move an object in front of the platform. See how it moves. And monitor the signal using Serial Monitor.

By: https://www.sparkfun.com/sparkfun-qwiic-mini-tof-imager-vl53l5cx.html

Practice 7 - Inertial Sensors

- Copy example InertialSensors.ino to your student folder,
  - ~WS_Zumo/Arduino/Examples/InertialSensors/InertialSensors.ino
  - Upload code and monitor the signals using Serial Plotter and Serial Monitor
  - Make sure to close Serial Plotter before uploading a new code.



Note: If you use Serial Plotter make sure to close it before uploading a new code

Exercise 4 - Tele Operate with sensor data

- Add sensor readings such as the IMU or TOF to the TeleOperate.ino code and send it to the Serial Monitor and Raspberry Pi.
- Modify the JoystickCamera.py example to plot the relevant sensor data in real time.
- The JoystickCamera.py example also records the readings to a recorded_messages.txt file.
    - Plot the recorded data using matlab or python.
- Turn off the system at the end of the session
    - Connect it to Power
    - Send shutdown command: “sudo shutdown -h now” or “zumoShutdown”
    - Force shutdown by pressing the On/Off Switch for 8 seconds
    - Turn off the power slide on the zumo Platform