ENGR 4421: Robotics II

Arduino

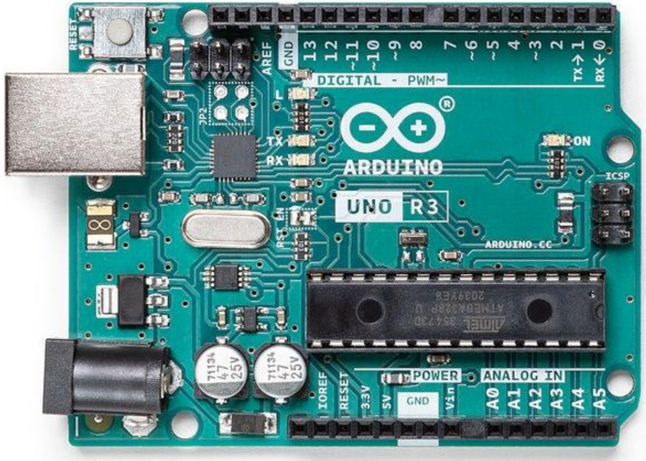
01/25/2022

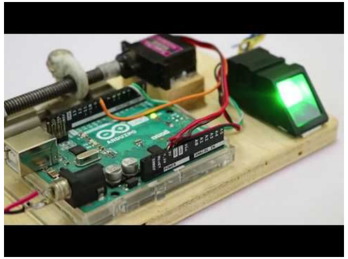
Outline

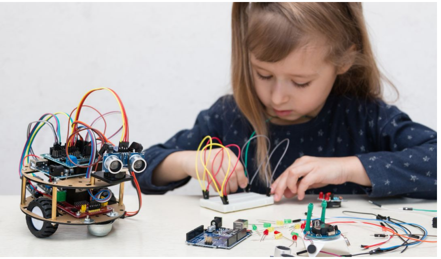
- Arduino Introduction
- Arduino Nano Every
- Get Started
- Blink LED
- Interrupts

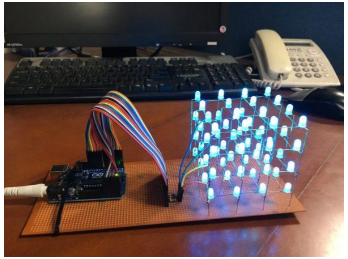
What is Arduino

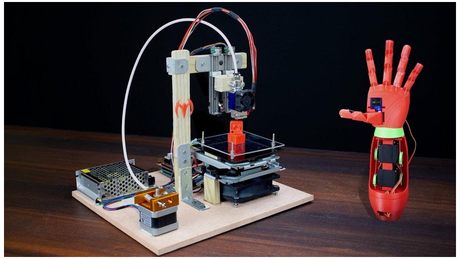
- Arduino is an open-source microcontroller project.
- Arduino is simple (OS not required).
- Easily build your project with Arduino.











Arduino Family

Arduino Uno

Arduino Leonardo

Arduino Due

Arduino Yún

Arduino Tre

Arduino Micro

Arduino Robot

Arduino Esplora

Arduino Mega ADK

Arduino Ethernet

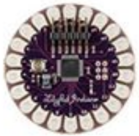
Arduino Mega 2560

Arduino Mini

LilyPad Arduino USB

LilyPad Arduino Simple

LilyPad Arduino SimpleSnap



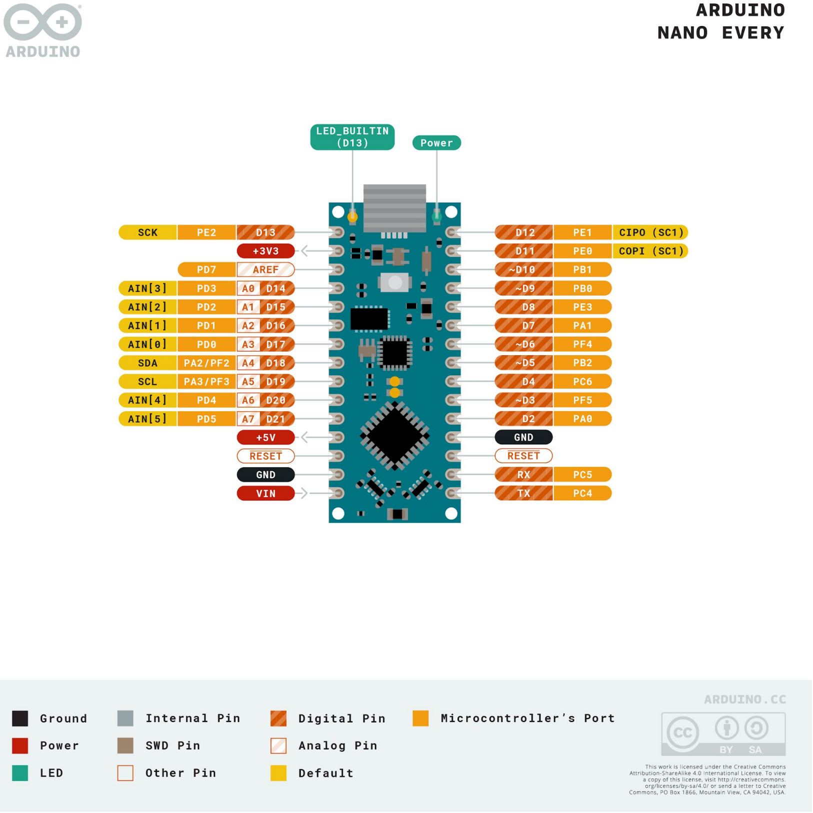
LilyPad Arduino

Arduino Nano

Arduino Pro Mini

Arduino Nano Every

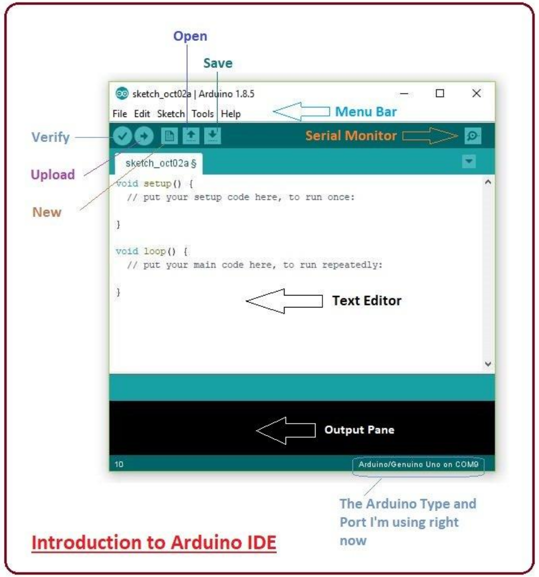
Microcontroller	ATMega4809 @ 16MHz
Digital I/O Pins	14
Analog Input Pins	8
PWM Pins	5 (3, 5, 6, 9 10 @ 976Hz)
I/O Voltage	5V



Get Started - Arduino Nano Every

1. Download and install Arduino IDE.
2. Install the megaAVR core.
    a. Open Arduino IDE.
    b. Navigate to **Tools > Board > Board Manage**.
    c. Search "**megaavr**", then install "**Arduino mbed-enabled boards**".
3. Select board: **Tools > Board > Arduino > Arduino megaAVR Boards**, select port: **Tools > Port**.
4. Open an example: **File > Examples > 01.Basics > Blink**, then upload.

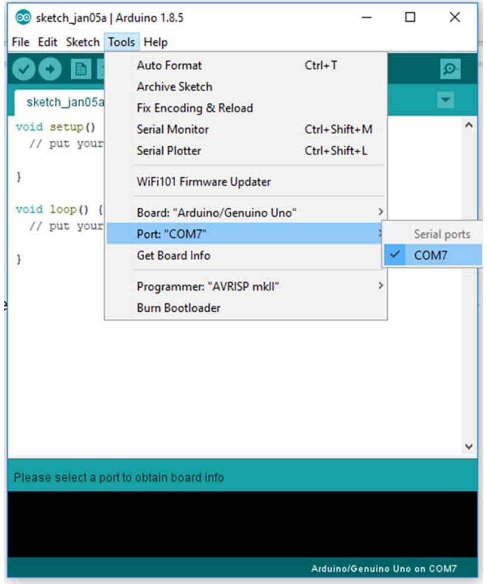
Arduino IDE



Open
Save
sketch_oct02a | Arduino 1.8.5
File Edit Sketch Tools Help
Menu Bar
Verify
Upload
New
Serial Monitor
sketch_oct02a §
Text Editor
Output Pane
10
Arduino/Genuino Uno on COM9
The Arduino Type and Port I'm using right now
Introduction to Arduino IDE

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Sketch



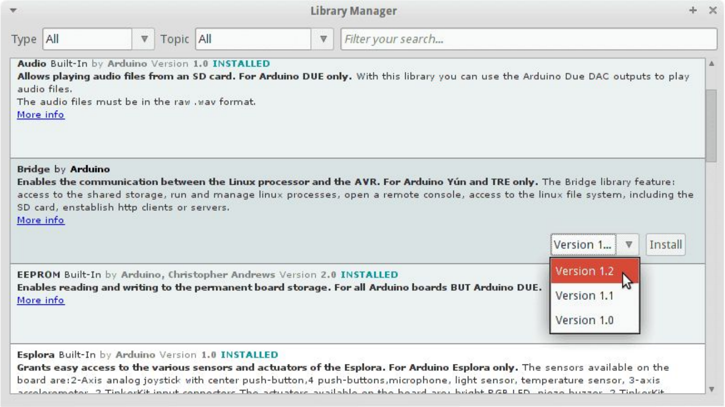
sketch_jan05a | Arduino 1.8.5
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM7"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ COM7
sketch_jan05a

```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Please select a port to obtain board info

Arduino/Genuino Uno on COM7

Select Board and Port



Audio Built-In by **Arduino** Version **1.0** **INSTALLED**
Allows playing audio files from an SD card. For Arduino DUE only. With this library you can use the Arduino Due DAC outputs to play audio files.
The audio files must be in the raw .wav format.
More info

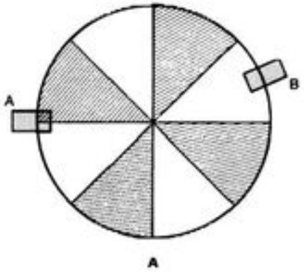
Bridge by **Arduino**
Enables the communication between the Linux processor and the AVR. For Arduino Yún and TRE only. The Bridge library feature: access to the shared storage, run and manage linux processes, open a remote console, access to the linux file system, including the SD card, enstablish http clients or servers.
More info

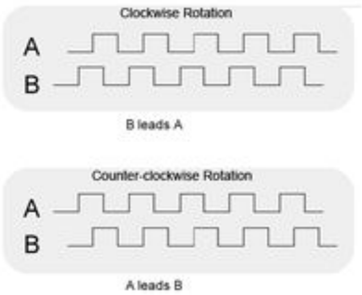
EEPROM Built-In by **Arduino, Christopher Andrews** Version **2.0** **INSTALLED**
Enables reading and writing to the permanent board storage. For all Arduino boards BUT Arduino DUE.
More info

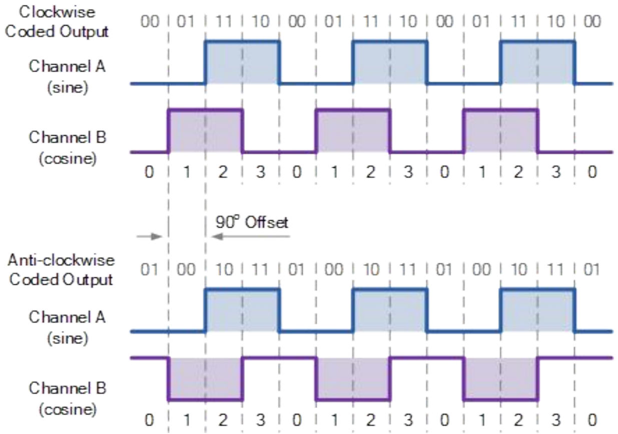
Esplora Built-In by **Arduino** Version **1.0** **INSTALLED**
Grants easy access to the various sensors and actuators of the Esplora. For Arduino Esplora only. The sensors available on the board are:2-Axis analog joystick with center push-button,4 push-buttons,microphone, light sensor, temperature sensor, 3-axis

Manage Libraries

Encoder Reading





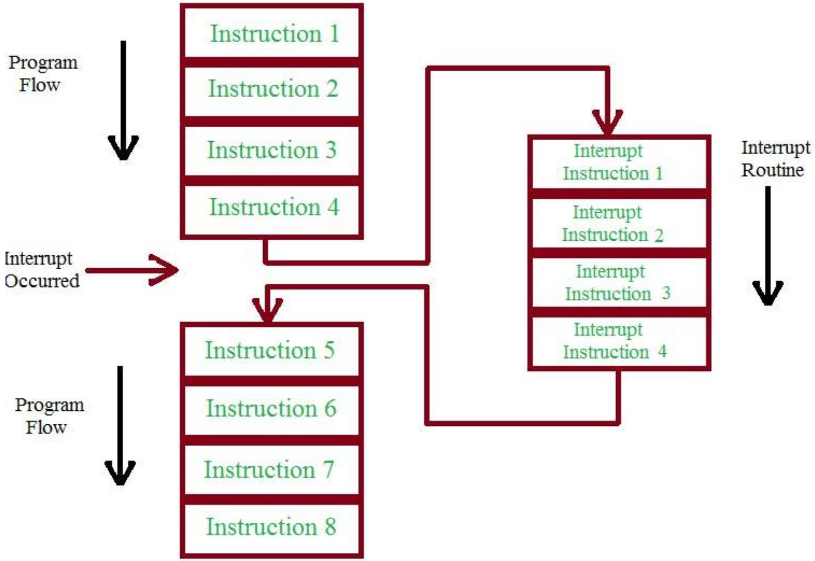


Cases:

- Clockwise: {0, 0} → {0, 1}; {0, 1} → {1, 1}; {1, 1} → {1, 0}; {1, 0} → {0, 0}
- Counterclockwise: {0,0} → {1,0}; {1, 0} → {1, 1}; {1, 1} → {0, 1}; {0, 1} → {0, 0}

Interrupts

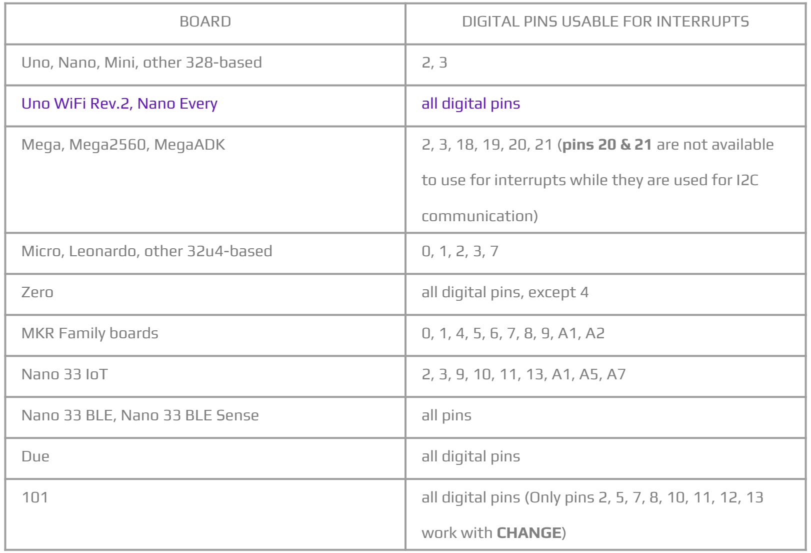
- Video tutorial: https://youtu.be/9VZUb5cMrV0
- Encoder reading example: https://dronebotworkshop.com/robot-car-with-speed-sensors/



```
const byte encA = 2;
const byte encB = 3;
int motorDir = 0;  // clockwise: 1, ccw: -1
long counts = 0;
int stateA = 0;
int stateB = 0;
int lastStateA = 0;
int lastStateB = 0;

void setup() {
  // setup pin modes
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(encA), ISR_updateA, CHANGE);
  attachInterrupt(digitalPinToInterrupt(encB), ISR_updateB, CHANGE);
  stateA = digitalRead(encA);
  stateB = digitalRead(encB);
  lastStateA = stateA;
  lastStateB = stateB;
}

void ISR_updateA() {
  lastStateA = stateA;
  stateA = !stateA;
  if (lastStateA == 0) {
    if (lastStateB == 0) {
      motorDir = 1; // last states: {0,0}, present states: {1,0}
    }
    else {
      motorDir = -1;  // last states: {0,1}, present states: {1,1}
    }
  }
  else {
    if (lastStateB == 0) {
      motorDir = -1; // last states: {1,0}, present states: {0,0}
    }
    else {
      motorDir = 1;  // last states: {1,1}, present states: {0,1}
    }
  }
  counts += motorDir;
  motorDir = 0;
}

void ISR_updateB() {
  lastStateB = stateB;
  stateB = !stateB;
  if (lastStateA == 0) {
    if (lastStateB == 0) {
      motorDir = -1; // last states: {0,0}, present states: {0,1}
    }
    else {
      motorDir = 1;  // last states: {0,1}, present states: {0,0}
    }
  }
  else {
    if (lastStateB == 0) {
      motorDir = 1; // last states: {1,0}, present states: {0,1}
    }
    else {
      motorDir = -1;  // last states: {1,1}, present states: {1,0}
    }
  }
  counts += motorDir;
  motorDir = 0;
}


void loop() {

//  counts += motorDir;
  Serial.println(counts);
}
```

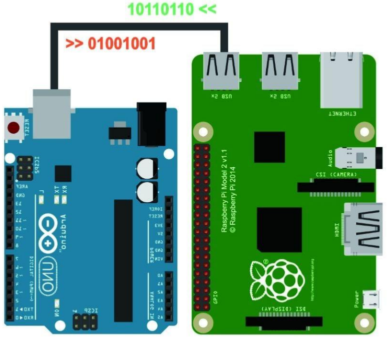
Interrupt Service Routine

- Interrupt types: LOW, HIGH, FALLING, RISING, CHANGE
- Needs to run quickly. Make sure the code running in ISR AS FAST/SHORT AS POSSIBLE.
- ISR function cannot take input parameters, or return values.
- Cannot pause in ISR (“delay(), millis() are disabled”).
- Only a certain number of pins are capable as “interrupt pins”.

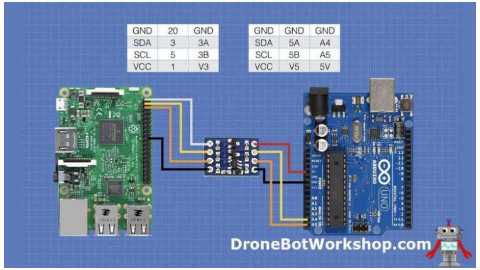


BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21 (**pins 20 & 21** are not available to use for interrupts while they are used for I2C communication)
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, A1, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with **CHANGE**)

Communication Protocols



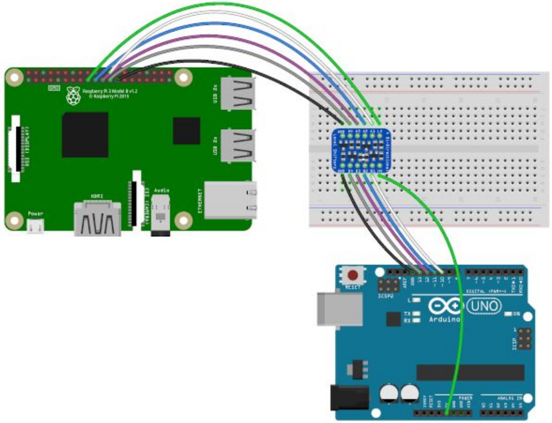
UART



GND	20	GND
SDA	3	3A
SCL	5	3B
VCC	1	V3

GND	GND	GND
SDA	5A	A4
SCL	5B	A5
VCC	V5	5V

I2C



SPI

Read Serial Data

```
#!/usr/bin/env python3
import serial

if __name__ == '__main__':
    ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
    ser.reset_input_buffer()

    while True:
        if ser.in_waiting > 0:
            line = ser.readline().decode('utf-8').rstrip()
            print(line)
```

More examples: https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/