ELEN30013 Workshop 2: Arduino

- For demonstrators and students: this workshop session will be marked during the workshop session. No report submission required.
- Attendance 20 points for on-time, 15 point for late (after 15 min), 5 point (after 30 min), 0 point (No show)

< Contents >

1. Introduction

This workshop will cover a basic introduction to Arduino programming and circuits. This workshop can be completed either online using Tinkercad or using an Arduino kit.

Tinkercad Circuits: https://www.tinkercad.com/learn/circuits

Arduino IDE: https://www.arduino.cc/en/software

By the end of the workshop, you should understand to write Arduino compatible code. You should have some knowledge of the main chip used on an Arduino Uno along with the circuit that the Arduino Uno is constructed from and some of the device limitations.

1.1. About Arduino Uno

An Arduino Uno R3 is an embedded development board for easy development. It can be programmed in C/C++ while making use of a number of functions that abstract away more low-level tasks such as manipulating registers. This is still possible, bare-metal programming will not be covered.

The core the Arduino Uno R3 is a ATmega328P chip that runs the code and a secondary chip to act as a USB to Serial bridge, nominally a ATmega16U. Both of these chips are from Atmel, that has been acquired by Microchip in 2016.

Use the following links to answer the following questions.

https://store.arduino.cc/usa/arduino-uno-rev3

https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

https://content.arduino.cc/assets/Pinout-UNOrev3_latest.pdf

Task 1. Solve below question sets (0 point)

1. What voltage levels are used for logic on an Arduino Uno?
2. What is the maximum current draw from each I/O pin can supply on an Arduino Uno?
3. What is the maximum current draw of the whole board?
4. What is the 5V voltage regulator IC part name on the Arduino Uno and what is the typical current it can regulate? (Hint: U1)

   https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf
5. What pins exist on the Arduino Uno and which are capable of Digital Read, Digital Write, Analog Read, Analog Write / PWM.

   https://content.arduino.cc/assets/Pinout-UNOrev3_latest.pdf

2. Arduino Programming

2.1. Bare minimum code

Read the bare minimum code and extended minimal code. There is no need to run the code they are simply to gain an understanding of the common code elements and the two core functions; setup and loop.

setup(): This function is run once on startup and is used to setup things such as serial and pin mode.

loop(): This function runs after the setup and once it is finished it runs again.

```
/*
    Title: Bare Minimum Arduino
    Description: A condensed bare minimum Arduino code template.

    The circuit:
    * No input
    * No output
    (Normally pin number and connections would be listed here)

    Created 10 05 2021
    By ESI 2021 Team
    Modified 11 05 2021
    By ESI 2021 Team
        - This is a multiline comment
*/
// This is a comment
// This code it run one at the start
```

```
void setup() {


}

void loop() {
 // This code is constantly rerun after the setup


}
```

```
/*
    Title: Minimal Arduino with notes on and references

    Description: A simple bare minimum Arduino reference file
    with sections appearing in the preferred and common order.

    The circuit:
    * No input
    * No output
    (Normally pin number and connections would be listed here)

    Created 10 05 2021
    By ESI 2021 Team
    Modified 11 05 2021
    By ESI 2021 Team
        - Note significant changes
*/

// Comments: https://www.arduino.cc/en/pmwiki.php?n=Reference/Comments
// Style: https://www.arduino.cc/en/Reference/StyleGuide
// Use indentation and spaces
/* Multiline
comment
*/

// Include:
// https://www.arduino.cc/reference/en/language/structure/further-syntax/include/
// #include <MyLibrarySearch.h>
```

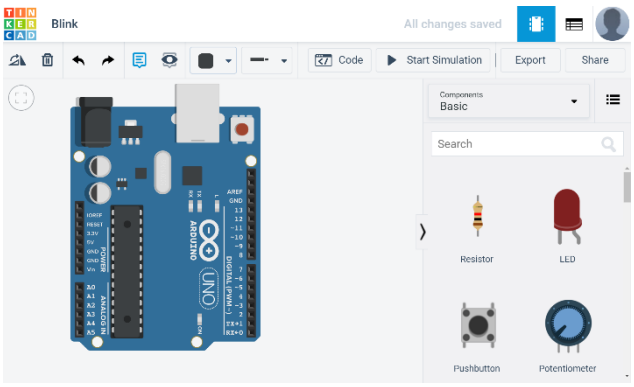
```
// #include "MyLibraryLocal.h"

// Defines:
// https://www.arduino.cc/reference/en/language/structure/further-syntax/define/
// NOTE: NO equal and NO semicolon
// #define ESI_IS_AWESOME true

// Variables: https://www.arduino.cc/reference/en/#variables
// int speed = 999;
// char driveLetter = 'T';
// char passcode[] = "ESI_2021!"
// NOTE: https://www.arduino.cc/reference/en/language/variables/data-types/string/

void setup() {
    // This code it run one at the start
}

void loop() {
    // This code is constantly rerun after the setup
}

// Functions: https://www.arduino.cc/en/Reference/FunctionDeclaration
```

Now we will use this knowledge of the code structure to make an LED Blink! Refer to the following steps, schematic and code. Use a 470 Ω resistor. Keep in mind that the LED's Cathode is connected to the short leg and the flat side. In this circuit the Cathode is connected to the ground.

Calculate the smallest resistor value assuming a Red LED is used with a forward voltage of 1.8V and has a current smaller than 16mA. These are very conservative numbers so verify that the 470 Ω resistor is larger than the calculated minimum.

Tinkercad

https://www.tinkercad.com/learn/circuits



After logging on to Tinkercad - circuits you will should have a page like the one above in your browser. Be careful to start a “circuit” and not a “3D Design”

On the right are a selection of components that can be dragged in. The code section will need to be switched to text.

The code editor will need to be changed to Text. There is a serial monitor at the bottom of the editor panel. If you cannot edit the code, make sure the simulation is stopped.

Shortcuts

R : Rotate

E or NumPad 5: hide / show editor code panel

C or NumPad 3: hide / show components panel

S : Start / Stop simulation

Ctl + Z : Undo

Ctl + Y : Redo

Arduino IDE

https://www.arduino.cc/en/software

After installing the Arduino IDE the program should open to a window like the one above.

Now connect your Arduino and set the port and board!

Tools > Board > Arduino Uno

Tools > Port > “Select the port with Uno in the name”

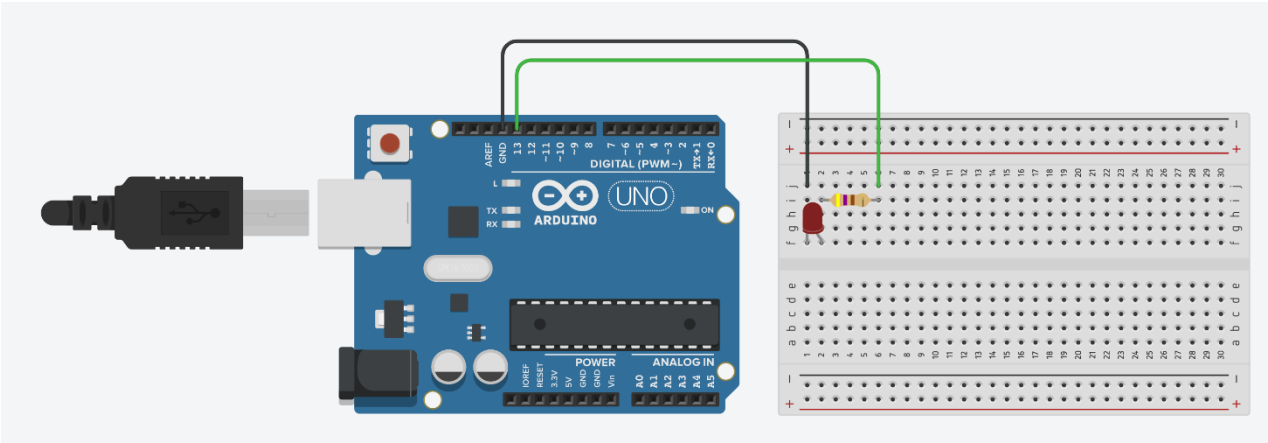
If the port is not immediately apparent you can check it with the device manager. Or disconnect the Arduino and see which COM port disappears, if there is no change there is likely a driver issue, faulty cable, or board issue. If possible try another restarting your computer or another USB port.

Some driver updates require a system restart as well. Depending on the board the CH340 driver may be needed: https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

Shortcuts

 Ctl + R : Verify

 Ctl + U : Upload




```
const int led_pin = 13;

void setup()
{
  pinMode(led_pin, OUTPUT);
}

void loop()
{
  digitalWrite(led_pin, HIGH);
  delay(1000);
  digitalWrite(led_pin, LOW);
  delay(1000);
}
```

Task 2. Solve questions below. (0 point)

1. In your own words how would you describe the elements of the bare minimum Arduino code? List any parts that you are uncertain of.
2. In your own words what does the function digitalWrite do and what inputs does it take?

   hint: try `digitalWrite(led_pin, 123);`
3. What units are delay in? Hint: just google
4. Briefly investigate and summarise in your own words what the following functions do:

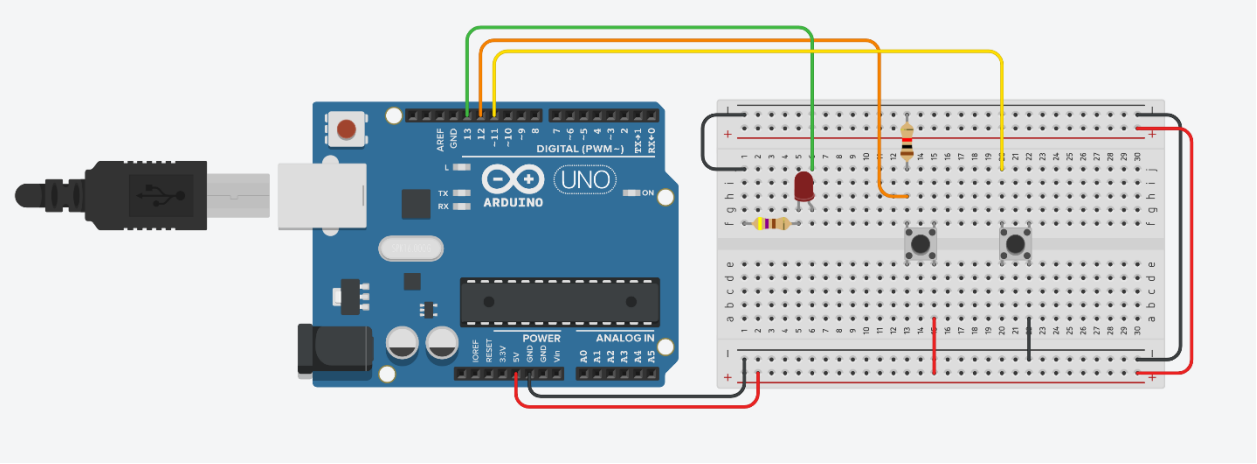
   `delayMicroseconds()`

   `millis()`

   `micros()`

2.2. Digital Read & Write

Task 3. In this exercise we will connect and program the Arduino to turn on an LED when a button is pressed. Use the given code to program the Arduino. (10 points)



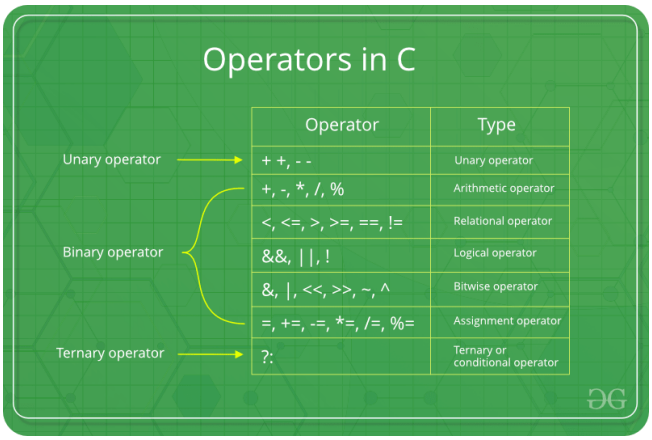

```
int data1 = 0;
int data2 = 0;
const int led_pin = 13;
const int btn1_pin = 12;
const int btn2_pin = 11;


void setup()
```

```
{
   pinMode(led_pin, OUTPUT);
   pinMode(btn1_pin, INPUT);
   pinMode(btn2_pin, INPUT_PULLUP);
}

void loop()
{
   data1= digitalRead(btn1_pin);
   data2= !digitalRead(btn2_pin);
   digitalWrite(led_pin, data1|data2);
   delay(10);
}
```

Task 4. Implement the system and then modify the code so that the LED will stay on (after releasing the button) for the duration that the button was pressed for. (10 Points)

Task 5. Solve below questions (0 points)

1. What is the difference between `INPUT` and `INPUT_PULLUP` ?
2. List 4 logical operators can be used in C/C++?

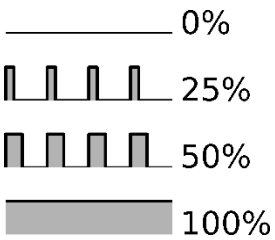


Reference: https://www.geeksforgeeks.org/operators-c-c/

3. What is the data size and range of values int, double, bool, and long can take?

2.3. Analog Read & Write

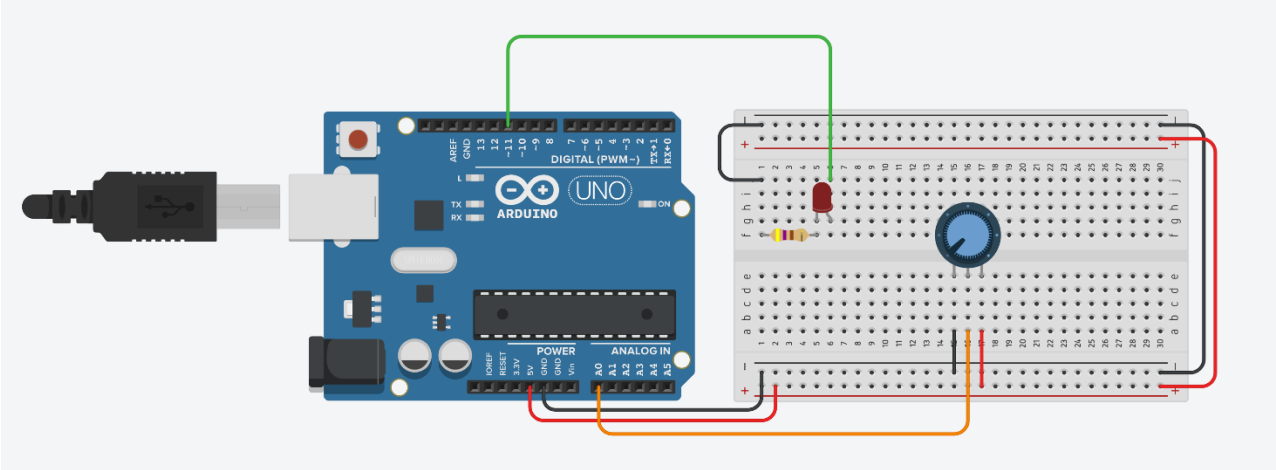
The Arduino is only capable of reading analog voltages on the Analog pins. It does this using an Analog to Digitial Converter (ADC) to represent the voltages between 0V and 5V (or AREF can be used) as a binary number of some size depending on the specific device.



The Arduino emulates an analog voltage by turning a pin on and off repeatedly at a certain frequency. The pins that are capable of this are denoted with a tilde ~. This type of modulation is called Pulse Width Modulation. The ratio of the on time to the on off cycle duration is referred to the duty cycle (%).

Task 6 (20 points)

Implement the following design using the code and schematic using a 1K potentiometer. Then modify the code so that the LED is brightest in the middle and a linear mapping to off at the far left and right of the potentiometer. Hint: the abs() function may be helpful.



```
int data_in = 0;
int data_out = 0;
const int led_pin = 11;
const int pot_pin = A0;

void setup()
{
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop()
{
data_in = analogRead(pot_pin);
            data_out = map(data_in, 0, 1023, 0, 255);
analogWrite(led_pin, data_out);
            delay(10);
}
```

Task 7. Aswer questions below (0 point)

1. How many bits is the Arduin Uno's ADC?
2. What frequency is the PWM output on the Arduino Uno?
3. How does the map function work?

2.4. Serial Read & Write

Serial is a very useful tool to transfer data from an Arduino to a computer or another device in real time. This can be highly beneficial for debugging or for expanding your projects!

Serial can run at various speeds and the Arduino supports baud rates of; 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, and 115200 bps. The most common speeds are 9600 and 115200 bps.

It is important to note that double quotes are used for strings and single quotes are used for characters.

Tinkercad the serial monitor can be found at the bottom of the editor. It uses a set baud rate of 9600 bps.

Arduino IDE the serial monitor can be found under Tools > Serial monitor (shortcut control + shift + M).

Matlab a serial port can be connected to and read with the following code using COM12 as an example port:

```
device = serialport("COM12",9600)
text = readline(device)
```

There are also read, write and flush commands.

A similar system as Matlab is used in Python using the Pyserial Library with import serial.

Task 8 (20 Points)

Using any of the previously mentioned methods load the following code onto the Arduino and interface with the serial port to turn on and off the inbuilt LED (pin 13)

Optional Task (5 points)

As a final exercise upload and play a game of Tic-Tac-Toe and improve one section of the code or add a hardware component.

```
#define BOARD_WIDTH 3
#define BOARD_LENGTH pow(BOARD_WIDTH,2)

int board[9];

const char blank_tile = '*';
const char user_tile = 'X';
const char comp_tile = 'O';

const int blank_id = 0;
const int user_id = 1;
const int comp_id = 2;

int player_move_text;

bool valid_move;
```

```
bool game_on = true;

// Array for board
// 0 1 2
// 3 4 5
// 6 7 8

void setup() {
  Serial.begin(9600);
  startGame();

}

void loop() {

  if (game_on) {
    if (Serial.available()&(remainingTurns()>0)) {
      player_move_text = Serial.parseInt() - 1;
      Serial.println(player_move_text);
      if ((remainingTurns() % 2) == (user_id % 2)) {

        valid_move = turn(user_id, player_move_text);
        if (!valid_move) {
          Serial.println("invalid move...");
          printBoard();
        }
      }
      checkWinner(user_id);
    }

    if (((remainingTurns() % 2) == (comp_id % 2)&(remainingTurns()>0))) {
      valid_move = false;
      while (!valid_move) {
        valid_move = turn(comp_id, random(BOARD_LENGTH));
```

```
    }
    printBoard();
    checkWinner(comp_id);

  }
  if ((remainingTurns() == 0) & (game_on)) {
    Serial.println("It was a tie!?!?!");
  }
 }
 else {
  startGame();
  game_on = true;
 }
 delay(100);
}



void printBoard() {
 Serial.println();
 for (int i = 0; i < BOARD_LENGTH; i++) {
  Serial.print((board[i] == 0) ? blank_tile : (board[i] == 1) ? user_tile : (board[i] == 2) ? comp_tile : '?');
  Serial.print(" ");
  if ((i % BOARD_WIDTH) == (BOARD_WIDTH - 1)) {
    Serial.print("\n");
  }
 }
 Serial.println();
}

void clearBoard() {
 for (int i = 0; i < BOARD_LENGTH; i++) {
  board[i] = 0;
 }
}
```

```
bool checkWinner(int player_id) {
  //for every for any row , col or diagonal and the value is eqaual to player_id
  bool is_winner = false;
  // Array for board
  // 0 1 2
  // 3 4 5
  // 6 7 8
  //Rows
  is_winner |= (player_id == board[0]) & (board[0] == board[1]) & (board[1] == board[2]);
  is_winner |= (player_id == board[3]) & (board[3] == board[4]) & (board[4] == board[5]);
  is_winner |= (player_id == board[6]) & (board[6] == board[7]) & (board[7] == board[8]);
  //Cols
  is_winner |= (player_id == board[0]) & (board[0] == board[3]) & (board[3] == board[6]);
  is_winner |= (player_id == board[1]) & (board[1] == board[4]) & (board[4] == board[7]);
  is_winner |= (player_id == board[2]) & (board[2] == board[5]) & (board[5] == board[8]);
  //Diag
  is_winner |= (player_id == board[0]) & (board[0] == board[4]) & (board[4] == board[8]);
  is_winner |= (player_id == board[2]) & (board[2] == board[4]) & (board[4] == board[6]);

  if (is_winner) {
    if (player_id == user_id) {
      Serial.println("YOU WIN!!!!!!!!!!!!!!!!!!!!!!!");
    }
    else {
      Serial.println("YOU LOSE.............");
    }
    game_on = false;
  }
  return is_winner;
}

int remainingTurns() {
  int turns_rem = 0;
```

```
  for (int i = 0; i < BOARD_LENGTH; i++) {
    turns_rem += (board[i] == 0);
  }
  return turns_rem;
}

bool turn(int player_id, int player_move) {
  bool is_valid_turn = false;
  if ( (player_move >= 0) & (player_move < BOARD_LENGTH)) {
    if (board[player_move] == blank_id) {
      is_valid_turn = true;
      board[player_move] = player_id;
    }
  }
  return is_valid_turn;
}

void startGame() {
  clearBoard();
  Serial.println("-------------------------");
  Serial.println("TicTacToe Game starting...");
  Serial.print("You will be playing as ");
  Serial.print(user_tile);
  Serial.println(" against the Arduino!\nInput a number to make your move.");
  Serial.println("1 2 3\n4 5 6\n7 8 9");
  printBoard();
}
```