Beginning With Intel Galileo Board

Pharino Chum[1*] **and Thavorac Chun**[2†§]

[1]Department of Electricala and Energy Engineering; [2]Department of Information and Communication Engineering, Institute of Technology of Cambodia

***For correspondence:**
pharino.chum@itc.edu.kh (Mr.);
thavorak@itc.edu.kh (Mr.)

[†]These authors contributed equally to this work
[‡]These authors also contributed equally to this work

Abstract In this training we focus on buidling your basic skill related to using Intel Galileo board, which is an Arduino supported board for leaner and innovator. In this training, we focus on basic knowledge of input/output. To make the learning intuative as much as posible, ww will work on smaller projects towards bigger and bigger projects.
The first part of this training, we will deal with digitial input and output hardware such as a LED and a push button. To make it more interesting, we will lern also the serial commincation, which will help you to troubleshoot your coding. Second part of this training will deal with reading analog input and generating analog signal to control a speed of a DC motor.
Uppon finish this training, you team are required to realized a project based on existed material and submit a short report to your course instructor within a week.

Project 1: Blinking an LED

In this activity, students will connect up the Intel Galileo to a simple circuit and make an LED blink using Arduino code.

Material

- 1x LED
- 1x 330 Ω resistor
- 1x Breadboard
- 1x Intel Galileo board
- 1x PC with Arduino IDE 1.6.4 (or above) installed
- 1x micro usb cable
- Some jumber wires.

Wiring

LEDs have a negative and a positive lead and current only flows one way. The convention is that the shorter lead is the negative lead while the longer lead is the positive lead. Please refer to figure (1) for detial wiring diagram.

- Connect the negative (shorter) lead of the LED to a resistor by twisting them together and connect the other end of the resistor to GND next to Pin 13.
- Connect the positive (longer) lead of the LED to Pin 13. Make sure there is no touching between the resistor and the positive lead of the LED.

Coding

From the Arduino IDE, go to **File** → **New** to create a new sketch for code. Be sure to select right board and port before uploading to target board.

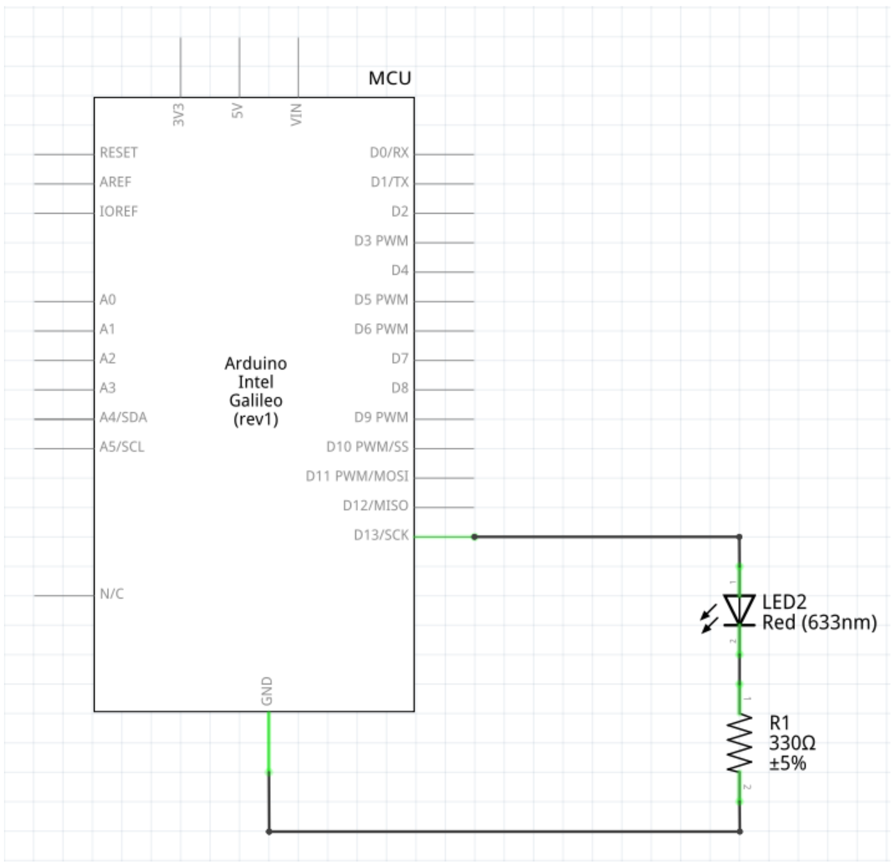

Figure 1. Blinking LED wire schema diagram.

```
/* Program for blinking a LED
 Description: this program will tickle a LED connected pin 13 for every second.
*/
// built-ing LED pin 13
int LED = 13;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED, OUTPUT); // set up LED pin as a output
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED, HIGH);  // turn LED on (HIGH voltage level)
  delay(1000);              // wait for 1 second
  digitalWrite(LED, LOW);  // turn LED off (LOW voltage level)
  delay(1000);              // wait for 1 second
}
```

Explanation of the Code

In Arduino programs, there are always two functions: *setup* and *loop*. The *setup()* function runs once when you load a new program or press the reset button. It is represented by the code:

```
void setup() {
```

```
....

}
```

The *loop()* function runs after the setup function and runs over and over again forever. It is represented by the code:

```
void loop() {
....

}
```

The digital pins on the Galileo, (0 – 13) located next to the USB slot can be set to be either output or input pins using the command, *pinMode()*. The *pinMode* function expects two inputs inside the parentheses: the first input is the number of the Galileo pin while the second input is either *OUTPUT* or *INPUT*.

In order to set an output on the Galileo we use the command *digitalWrite*. It also takes two inputs: the first input is the number of the Arduino pin to write to and the second input is whether to write *HIGH* or *LOW*.

The delay command makes the program wait for a certain number of milliseconds before moving onto the next command. Thus:

```
delay(1000);
```

instructs the code to wait for 1 second.

Comments are lines of script inside an Arduino sketch that provide notes to yourself or other users. These can be about the program's functionality, any issues with the code, or anything else you would like a user to understand about your program. Comments are ignored by the Arduino IDE and are not uploaded to the Intel Galileo hardware. Comments are often signified with the symbols "//" or "/* */" and can be written as follows:

```
// This is a single line comment. Anything after the slashes is a comment
// to the end of the line
/* this is a multiline comment
– use it to comment out whole blocks of code
*/
```

Project 2: Push Button

This project is an example of how we can read digital (1 or 0) information from source such as a switch etc. For this project, in order to turn on the LED, user will need to press a button. The LED will stay on as long as user keeps pressing the button only.

Material

- 1x LED
- 1x Push button
- 1x 330 Ω, 1x $1K\Omega$ and 1x $10K\Omega$ resistors
- 1x Breadboard
- 1x Intel Galileo board
- 1x PC with Arduino IDE 1.6.4 (or above) installed
- 1x micro usb cable
- Some jumber wires.

Wiring

We use the same wiring from previous project. We add some component as showing in figure (2). Please refer to figure (2) for detial wiring diagram.

- Connect one end of $1K\Omega$ resistor to voltage source 5V on board Galileo. And Connect another end to pushbutton.
- Connect output of pushbutton to pin 12 which is near pin 13 of board. And also connect this pin to resistor $10K\Omega$.
- Connect output of resistor $10K\Omega$ to GND.

It is notice that when button is open (no-press), the voltage of pin-12 is 0V which corresponds to logical state 0. While it is pressed, the voltage is $5 \times (10K)/(10K + 1K) = 4.545V$ wich corresponds logical state 1.

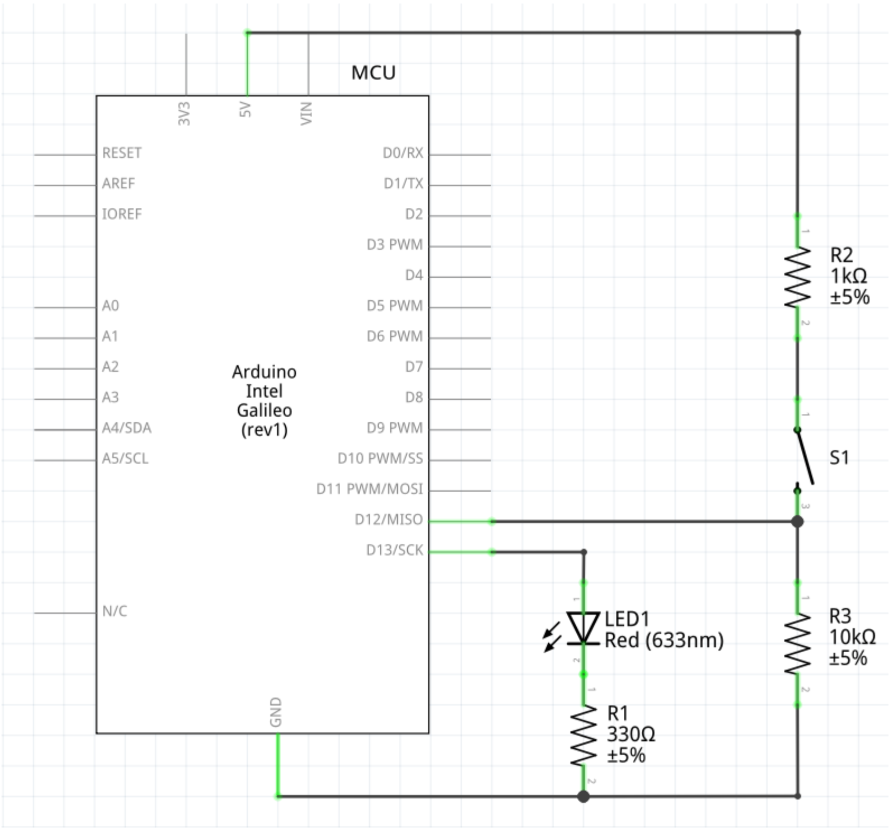

Figure 2. Project 2 wiring diagram.

Coding

From the Arduino IDE, go to **File** → **New** to create a new sketch for code. Be sure to select right board and port before uploading to target board.

```
/*
 *  Program:  push  button
 *  Description:
 *  Input:  Push  button  (pin-12)
 *  Ouput:  LED  (pin-13)
*/
```

```
// I/O declaration
int led = 13;
int button = 12;
int buttonState = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode(led, OUTPUT);
  pinMode(button, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  buttonState = digitalRead(button); // read state of button
  if(buttonState)
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);
}
```

Explanation of the Code

In this sketch, *digitalRead()* function reads the value from a specified digital pin, either *HIGH* or *LOW*.

Project 3: Serial Communication

This project is an example of how we can use serial communication to pass data from Galileo board to host computer via USB interface. To demostrate this, board must count number from 0 to 10 for every second and send those number to host computer via a serial terminal.

Material

- 1x Intel Galileo board
- 1x PC with Arduino IDE 1.6.4 (or above) installed
- 1x micro usb cable
- Some jumber wires.

Wiring

There is no need for wiring for this project. Just a simple setup of Galileo board to computer via USB cable.

Coding

From the Arduino IDE, go to **File** → **New** to create a new sketch for code. Be sure to select right board and port before uploading to target board. To open serial monitoring program, go to **Tool** → **Serial Monitor**.

```
/*
 * Program: serial
 * Description:
 * Input: NA
 * Ouput: NA
*/
// variable declaration
```

```
int count = 0; // to store counting value

void setup() {
  // setup serial communication
  Serial.begin(9600); // open the serial port at 9600 bps:

}

void loop() {
  // put your main code here, to run repeatedly:
  // start send data to host computer
  Serial.print("Start␣serial␣communication"); // send string
  Serial.println();                           // enter a new line

  while(count <= 10)
  {
    Serial.print("␣>>␣counting␣value␣=␣");  // print string
    Serial.print(count, DEC);                // print count as decimal number
    Serial.println();
    count ++;
    delay(1000);                             // wait for 1 second
    }
  // finish counting to 10, reset count to 0
  count = 0;
}
```

Project 4: Let’s Push and Count!

Using wiring diagram in project 2, develp a sketch for Galileo board to be able to count number of each time user push a button. Also, each time button is pushed, it need to send counting number to host computer via a serial terminal. When counting reach value of 10, turn on LED for 3 second and then turn LED off. Then restart the program.

Acknowledgments

Additional information can be given in the template, such as to not include funder information in the acknowledgments section.

References

Sarah Boyd DG John Burfoot, Howe C. A Teacher’s Guide to the Intel® GALILEO. . 2015; http://www.macict.edu.au/wp-content/uploads/A-Teachers-Guide-to-the-Intel-Galileo-Final.pdf.