Embedded Hardware Design

Monsoon 2017

Topic: Basics of Arduino Programming Language

Please learn the basics of Arduino language before attending the upcoming lab and try to translate your AVR C codes into Arduino

In the previous two labs, we had used the AVR style of C coding to program the Arduino board. This was done in order to give you a taste of register level access in microcontroller programming, i.e, how the registers are utilized behind the scenes to actually make your program work. C is still the most widely used programming language for embedded processors/controllers in the industry. The Arduino, however, provided an abstraction layer on top of C to minimize the register level interaction of the user. This made things very simple for people who just wanted to make prototypes and projects. Just like how Python is much simpler than C and supports a vast number of modules, so is the case with Arduino. Open source libraries exist for all hardware modules which can be easily used in Arduino.

Program Structure-

The following two functions are necessary-

1. `void setup():` This function is used for initializations and setting direction (Input/Output) of the pin.
2. `Void loop():` Equivalent to while(1). It is an infinite loop where your main logic is written.

```
//Headers (if any)
//Global variable Declarations (if any)
void setup()
{
    ...  //Initializations, Pin directions, Serial, etc
}
void loop()
{
    ...  //Control Logic
}
```

Basic Functions-

1. **`pinMode(pin, mode):`** This function controls the DDRx register. Suppose, my LED is connected to the digital pin 5 of the Arduino board. To make it an output pin, I'll simply write the following in void setup()-

   ```
   void setup()
   {
     pinMode(5, OUTPUT);
     }
   ```

   Other modes supported by this function are INPUT and INPUT_PULLUP

2. **`digitalWrite(pin, value):`** This function controls the PORTx register. We can give HIGH or LOW to an OUTPUT pin. For example, to blink an LED-

   ```
   void loop()
   {
     digitalWrite(5, HIGH);
     delay(1000);
     digitalWrite(5, LOW);
     delay(1000);
     }
   ```

   The *setup()* from (1) and *loop()* from (2) is a complete code for blinking an LED!

3. **`digitalRead(pin):`** This function controls the PINx register. We can read data from any INPUT pin. It returns the value read. Suppose Pin 6 was set as input. Then-

   ```
   void loop()
   {
     int a = digitalRead(6);
     ......
     }
   ```

4. **`Serial functions:`** The default serial UART port exists on digital Pin1 and Pin0. You can use this to communicate with the serial monitor-

a) `Serial.begin(9600:` To start UART communication with the Computer at 9600 baud rate. This goes in the *void setup().*

b) `Serial.print(“”)` : Print a value/string to the other UART device.

c) `Serial.read()` : Read data from the Serial device. It returns the value which can be read in a variable.

If-Else and Loops-

The if-else syntax is exactly same as the C coding.

Arduino, just like C, supports 3 loops-

a) *for* loop

b) *while* loop

c) *do-while* loop

Functions in Arduino-

Just like C, you can define your own functions in Arduino with the same syntax. For example, to multiply two numbers and return their product-

```
int Multiply(int x, int y){
    int pro;
    pro = x*y;
    return pro;
}
```

This has to be written outside the loop() and setup(). It can be called from inside the loop().

Resources to Study-

Arduino has a huge community online. You can find answers to all your questions just by *googling!* You can follow the famous Arduino tutorial series by Jeremy Blum on YouTube: https://www.youtube.com/playlist?list=PLA567CE235D39FA84

This doc is also pretty useful-
http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf