whitehomebuttonTempleT_smElectrical and Computer Engineering

## EE3613 Processor Systems Laboratory

## 

## LAB FINAL PROJECT

## DC Motor Control and Digital Speedometer

## Fall 2019

## OBJECTIVES

* To learn how to operate IO port of ATmega324PB Xplained Pro board using C
* To learn how to generate PWM using AVR timer
* To learn how to control DC motor speed
* To learn how to utilize interrupt functionality of the processor
* To learn how to measure the duty cycle and calculate RPM of the motor

## REFERENCES

Mazidi and Naimi, “The AVR Microcontroller and Embedded Systems,” Chapter 7,8,9 and 10

## MATERIALS

Atmel Studio 7, ATmega324PB Xplained Pro board, DC Motor, OLED Display, Optical interrupter

## BACKGROUND INFORMATION

***PULSE WIDTH MODULATION:***

In this lab your code will generate spin of motor using the port and pin operation. The bit operation will create pulse width modulation (PWM) signal by toggling an output port (Port E bit 2) between 0 (0V) and 1 (3.3V). PWM is a very popular method to control speed of dc motor, brightness of LED, temperature of heating elements, etc. A PWM signal consists of two main components: Frequency and Duty cycle. The frequency determines how fast the PWM completes a cycle (i.e. 1000 Hz would be 1000 cycles per second), and therefore how fast it switches between high and low states. The duty cycle describes the amount of time the signal is in a high (on) state as a percentage of the total time of it takes to complete one cycle. By cycling a digital signal off and on at a fast-enough rate, and with a certain duty cycle, the output will appear to behave like a constant voltage analog signal when providing power to devices.

**Example:** To create an average 3V output, given a power source that can be either high (on) at 12V, or low (off) at 0V, you can use PWM with a duty cycle of 25% which outputs 12V 25% of the time. If the digital signal is cycled fast enough, then the voltage seen at the output appears to be the average voltage. If the digital low is 0V (which is usually the case) then the average voltage can be calculated by taking the digital high voltage multiplied by the duty cycle, or 12V x 0.25 = 3V. Selecting 50% duty cycle would yield 6V, 75% would yield 9V, and so on. The PWM is adjustable between 0-100% duty cycle. Remember, rotation of the DC motor is directly related to the applied DC voltage.

## 

## Figure 1: 0%, 25%, 50%, 75% and 100% Duty Cycle Examples

***INTERRUPTS***

## An interrupt is a condition that causes the microprocessor to temporarily work on a different task, and then later return to its previous task. Interrupts can be internal or external. Internal interrupts, or "software interrupts," are triggered by a software instruction and operate similarly to a jump or branch instruction. An external interrupt, or a "hardware interrupt," is caused by an external hardware module. The processor stops what it is doing, it reads the input from the buttons or digital signal input port, and then it returns to the current program. You may use interrupt for reading signal from push button or reading pulses from the opto-interrupter.

## https://www.edsim51.com/8051Notes/8051/images/programFlowWithInterrupts.GIF

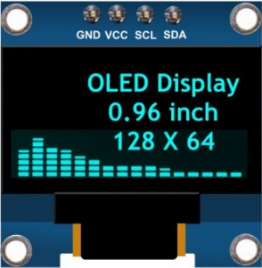
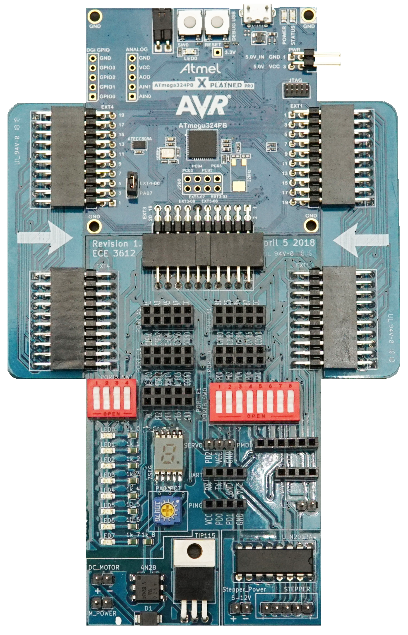
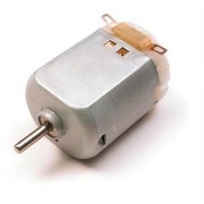
**Figure 2:** Graphical representation of ISR and main program

***SPEEDOMETER***

We used an opto-isolator (optocoupler) to separate the motor power circuits from the ATmega324PB to help lessen the likelihood of blowing something up (Figure: 4). A device similar to an opto-isolator is an opto-interrupter which has an air channel between the IR light emitting diode and the IR detector transistor, see Figure: 3d. An opaque object passing between the diode and the detector causes the transistor to turn off, resulting ‘interrupt’ event. We can tie the transistor to a I/O pin on the development board and detect the interruption. You will have to make a wheel with opaque and transparent or solid and hole, see examples in Figure: 3g. Make sure your wheel is properly balanced when you rotate it with a motor. If you rig up the motor base in a way that the wheel spins through the slot in the opto-interrupter, each time the optical window passes; the transistor turns on and back off. If we write our software so that a voltage change on the pin attached to the opto-interrupter cases an interrupt in the Atmel board, we can count those interrupts. If we count for exactly one second, we will have the number of slots that passed in one second and thus we can calculate the number of complete cycles per second.

For example, if there is N = 1 optical window in the disc and we have counted C = 60 pulses in one second, we can calculate the number of revolutions per minute (RPM).

**e**



ATmega 324PB Board

Photo Interrupter

DC Motor

OLED Display

Encoder Wheel with Station

9V Battery and connector

**a**

**b,c**

**d**

**f**

**g**

***REQUIRED ITEMS:***



9V

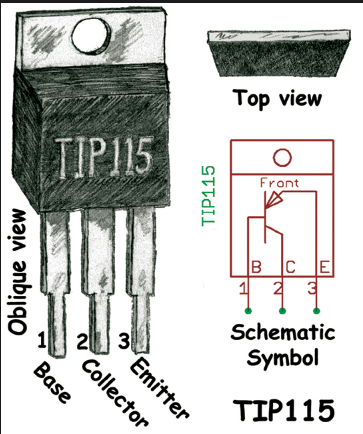
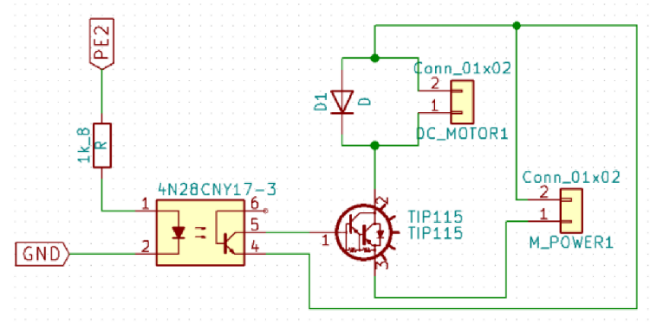
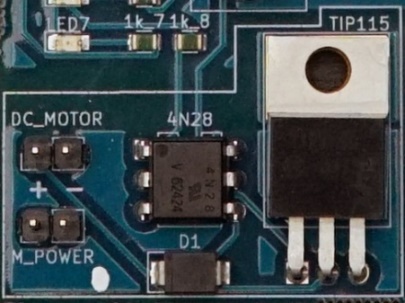
**Figure 3:** Some components that might be required for this project

## ACTIVITIES:

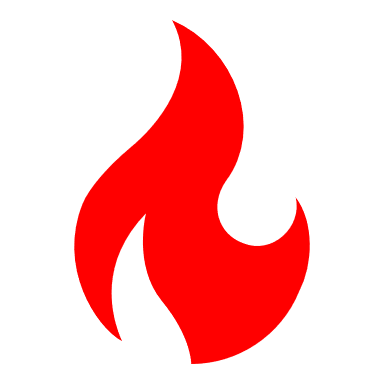
There are two activities for this final project. You are going to build a circuit to control DC motor and build a speedometer using optocoupler to measure the motor’s speed. Follow the steps for each activity.

**ACTIVITY 1 (Connections)**

Assemble your circuit with the ATmega board, a dc motor and battery. For this activity we will also use a push button or on-board switches and a photo interrupter. The DC motor will be connected to the designated **DC\_MOTOR** port of the board (Figure 4).



**Figure 4:** DCMotorcontroller circuit on the board and corresponding circuit diagram

******Note:** PE2 is connected to a 4N28 opto-isolator. This IC prevents noise and feedback from being introduced. A PNP Darlington transistor is used to amplify the signal. An external power source is required. You can use 9V battery and connector for this project (preferable).



9V

**DO NOT CONNECT THE BATTERY IN REVERSE POLARITY, IT WILL DAMAGE THE BOARD.**

For PWM input configuration control, you can use switch connected to PORTB for selecting the duty cycle and use PORTA to display the input (you can use 7-segment display or LEDs). For getting input, make sure you enable pull up resistor for that specific port bit in your code.

Top View

Emitter

Detector

A close up of a sign

Description automatically generated A picture containing object

Description automatically generated

**Important**: The 2 short legs of the senor are pin 2 and 4. You must ground them with the common ground of the circuit.

E

**4**

**3**

2

**1**

**S**

A screenshot of a cell phone

Description automatically generated

S

E

**3**

**4**

**2**

**1**

=2.2kΩ

**Vcc = 5V**

**Vcc = 5V**

**PD2**

**reading**

A circuit board

Description automatically generated

**150+150=300Ω**

Vcc=5V

**Pin 1**

**Pin 4**

**Pin 2**

**PD2 of the controller board**

Ground

Ground

**2.2kΩ**

**Pin 3**

Vcc=5VV

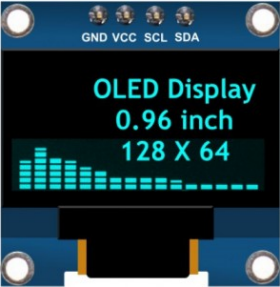
S

E

**Figure 5:** Opto-interrupter IC pin diagram, internal circuit diagram, and physical connections

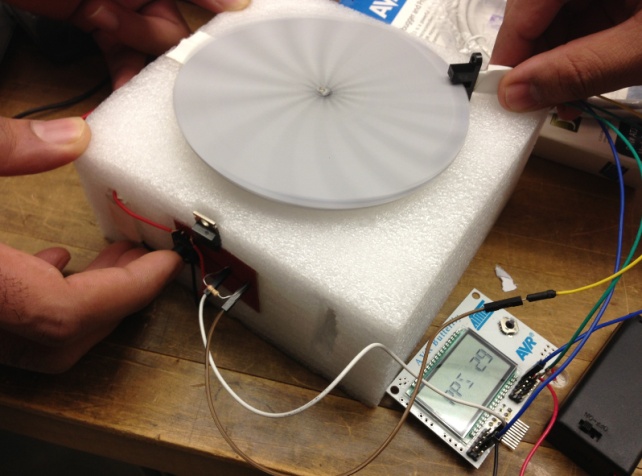
Now, you need to connect the opto-interrupter IC to the board. You must connect the **Anode** pin with a **current limiting resistor** (RD=150-1000 ohms) to prevent damaging the IR LED. The output of the IC can be connected to any input pin of the board similar to the push button switch. Make sure you enable pull up resistor for that specific port bit in your code. This is where you will utilize interrupt functionality to read the pulses.

The provided OLED I2C display can be used to display the RPM and PWM %, information. You can use previously supplied example codes to learn how to operate this i2c display. Make sure to connect the appropriate pins to the I2C connector of the development board. For the current version of the board, you need to switch the SCL and SDA line to make the display work.



**Figure 7:** OLED display

Build a motor base that accommodates the motor wheel and the sensor. Images from a previous project is attached to show an example but try to be creative.

**** 

**Figure 8:** Motor wheel - stationary and spinning

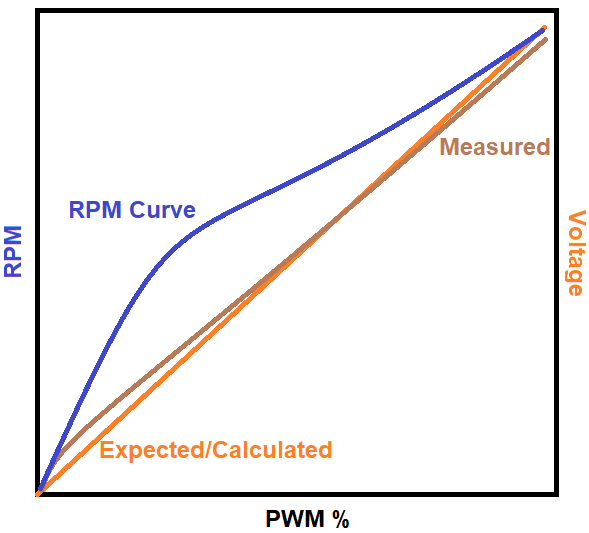
**ACTIVITY 2 (Coding and Measurements)**

Write a C code which generates different duty cycles of PWM. Send a 16kHz PWM signal to motor controller circuit to operate the DC motor. Apply different duty cycles (mode) using a push button or switches (see Table 1). Every time you change the mode, the PWM will change as following: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%. Monitor the changing RPM using the opto-interrupter sensor. Display the values using the display. The minimum information you are required to show on the display are: Mode (1-9), PWM%, RPM or RPS. You can add any other necessary or decorative / startup text/image you want in the display. Just make sure, the information is easily readable when you capture the video of your experiment.

Complete and include the following table in your project report with measured data. Measure the DC voltage using multimeter mode of the NI virtual bench. The RPM or RPS should be measured using your setup and code.

Table 1. Summary Table (You must include this table in the report)

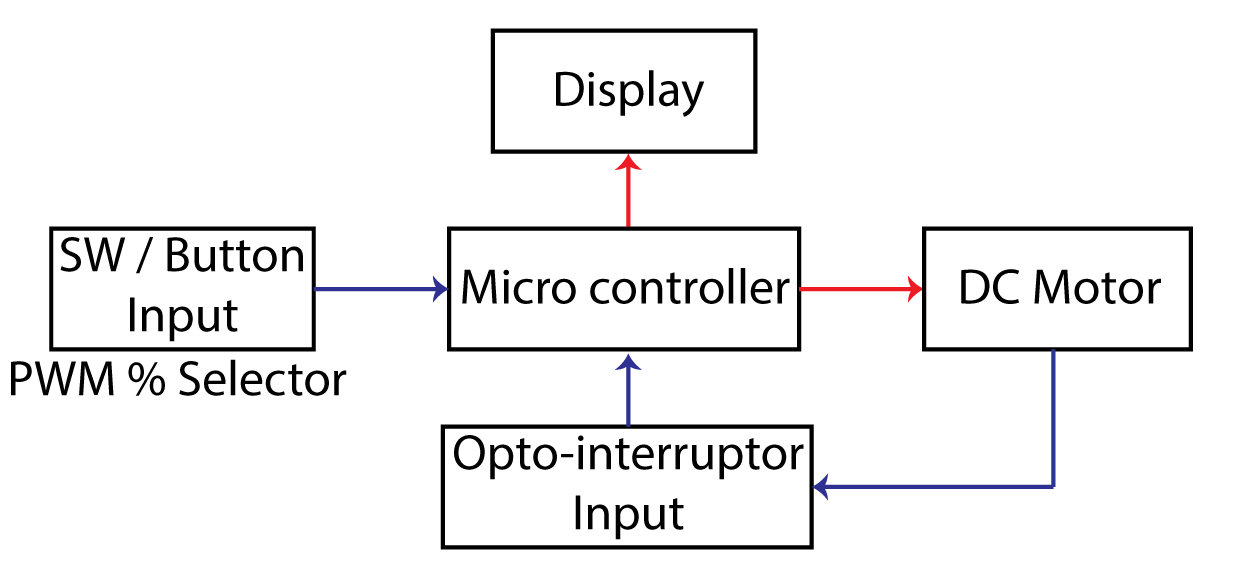
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Push Button or PB Switch**  **(bit7,6,5,4,3,2,1,0)** | **PWM Duty Cycle (%)** | **DC Voltage (Calculated)** | **DC Voltage Measured** (Across the Motor Connection) | **RPM or RPS** |
| PB Press 1 or 00000001 | 10 | ##.## V | ##.## V | ### |
| PB Press 2or 00000010 | 20 | ##.## V | ##.## V | ### |
| : | : | : | : | : |
| PB Press 9 or 00001001 | 90 | ##.## V | ##.## V | ### |

From the table, plot a graph to show the relationship between the motor’s RPM and their corresponding duty cycles. Also, make another plot to show the relationship between the voltage across the motor (both calculated and measured) and the duty cycles.

**GRAPH 1:** OUTPUT VOLTAGE vs. Duty Cycle

**GRAPH 2:** RPM vs. Duty Cycle

**Operation Summary:**



## REPORT

You must submit your report in doc and pdf format. Also, your codes must be saved and submitted in a .txt file format. Submit all three files to the LAB final report submission assignment. The lab report must be submitted by December 12, 2019, 11:59 pm (section 1 and 2 both).

IMPORTANT: DO NOT COPY FROM LAB MANUAL OR ANY OTHER SOURCES >>

**ECE3613 Processor System Laboratory Rubric**

**Lab Final**

**Section: 001 / 002**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Report Section** | **Part** | **Contents** | **Full Points** | **Earned Points** | **Comment** |
| **Cover Page** |  |  | 5 |  | Lab report Cover page |
| **Introduction** |  |  | 15 |  | Objective (5pts) and background information(10) |
| **Procedure** |  | Flowchart | 10 |  |  |
|  |  | Method | 20 |  | * Complete description of your code design and key code section explanation (breakdown each section and explain) –include important instructions and functions such as timer, interrupt, port operations so on.   **Note: Do not put the entire code in here.** |
|  |  | Hardware Design | 20 |  | * Complete description of your hardware system design and components specification |
| **Result** | I | Result | 30 |  | * Push Button or Switch control result to generate different duty cycles (%), use oscilloscope reading for the duty cycles (10pts) * DC motor control using PWM and voltage output measurement (20pt) – you must include the graph 1: **Duty Cycles vs. Voltage** |
|  | II | Result | 30 |  | * Optical Sensor Reading to count the revolutions for DC motor and computation for RPM – you must include the graph 2: **Duty Cycles vs. RPM** (20pts) * OLED readings for the RPM or RPS – select 3 examples of duty cycles and include the pictures of the reading (10pts) |
|  | III | Result | 20 |  | * Summary table (Table 1) and full description of the table |
| **Discussion** |  | Evaluation | 10 |  | Result Verification and Analysis of the result I: Compare the expected values and readings (error % from the expected) |
|  |  | Evaluation | 10 |  | Result Verification and Analysis of the result II: Compare the expected values and readings (error % from the expected) |
| **Conclusion** |  |  | 10 |  | Summary of the lab final project |
| **Appendix** |  | Code | 10 |  | Full comments, text file submission |
| **Video Link** |  |  | 10 |  | Show the full operation – Include all duty cycles running operation with the switch operation |
| **Total** | | | **200** |  |  |