LAB REPORT 4: INTERFACING WITH THE WORLD ON THE MSP430FR2344

Written by: M. Pouya Mirzaei
Class: Embedded Biomedical Instrumentation (BIEN 4220) for Spring 2024
Instructor: Dr. Robert Scheidt PhD
Teaching Assistant: Nathan Balthazor
Deadline: March 21st, 2024 at 1:45 PM

INTRODUCTION

In this lab report, the essential concepts and practical applications of interfacing with external components using the MSP430FR2344 microcontroller were investigated. Throughout the project, key electronic components such as the transistor, servo motors, and stepper motors with H-Bridge assembly were explored. The transistor's role as an electrical switch is examined, highlighting its ability to amplify and control electrical signals. Additionally, servo motors are studied, focusing on their use of Pulse Width Modulation (PWM) to achieve precise control over movement within a specific range. Lastly, the challenges of controlling bi-polar stepper motors with an H-Bridge are navigated Through hands-on experimentation and data analysis, this report aims to provide a comprehensive understanding of interfacing techniques essential for embedded systems design.

MATERIALS AND METHODS

This lab overviewed the use of three separate tools: the transistor, the servo, and the stepper motor/H-bridge.

THE TRANSISTOR

For this section of the lab, the ZTX651 NPN transistor was used.

To start off, it was important to understand the basic values and outputs of the transistor. In order to do this, a basic NPN BJT Transistor circuit (Figure 1) was made using a triple-output power supply (V_{CC}), a function generator, a base resistor (R_B) and a load resistor (R_L). Along with this, a digital multimeter (DMM) was used to record measurements throughout the circuit. The triple output power supply was connected to the collector in series with a 1 kΩ resistor. Next, a function generator, set to DC output, was connected to the base of the transistor in series with a 10 kΩ resistor, called the base resistor (R_B). The triple-output power supply (V_{CC}) was set to approximately 12 Volts, and the function generator was offset to set place the minimum at 0 Volts.

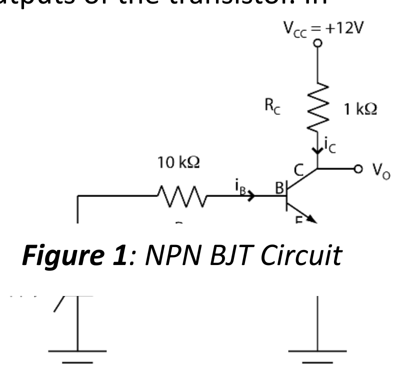


***Figure 1**: NPN BJT Circuit*

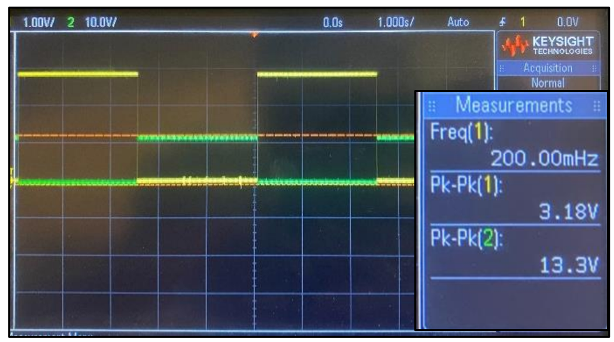
Once this setup was prepared, several measurements were made at different function generator outputs ranging from 0 to 6 Volts. These measurements were then recorded in **Table**

1 of the Results section. Along with this, the transistor current gain (β) was calculated using the formula below:

$$\beta = \frac{I_C}{I_B}$$

The average transistor current gain was also calculated and written in the Results section.

After characterizing the transistor, it was time to start exercising its use as an electrical switch. Using a 0 to 3V range, a square wave output was generated from the function generator. The oscilloscope was connected to both the input voltage (V_{IN}) and output voltage (V_O) of the circuit. <u>The relation between these two voltages can be seen in</u> **<u>Figure 2</u>**<u>.</u> From this figure, it is clear to see that the output voltage (13.3 V) is significantly larger than that of the input voltage (3.18 V), showing the use of the transistor as a switch where a small input voltage can be used to control a larger output voltage load.



<u>Figure 2</u>*<u>: a scope screen shot with the input/output traces identified and amplitudes indicated.</u>*

Now that the relation between input and output has been clarified. It was time to use this switch system to control a DC motor. To do this, the 1kΩ load resistor was replaced with a DC motor and a fly-back diode as shown in Figure 3. The purpose of the fly-back diode is to provide a clean transition for the voltage as the motor is turned off and on.

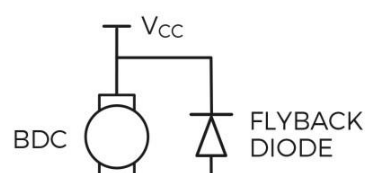
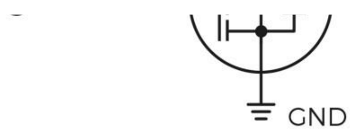


Figure 3*: The same circuit as before, but with a DC motor and flyback diode replacing the Load Resistor.*



With this <u>the frequency and duty cycle were modulated.</u> Using a 50% duty cycle, when the period was long, the motor would turn on and off consecutively. However, as the period shortened, the momentum of the motor would continue to the next cycle, making the motor appear to turn at a slower rate rather than being turned off and on. Using a constant period (especially when using a shorter constant period) if the duty cycle percentage was increased, the motor would appear to run faster. Alternatively, if the duty cycle percentage was decreased, the motor would appear to run slower. Hence, if one was to develop a vehicle, the speed of the tires would be a good indicator of the duty cycle of the motor. The greater the speed, the higher the duty cycle percentage and, slightly less likely, the shorter the period.

A video of the motor reacting with a period of 0.5 Hz and duty cycle of 25% was taken and attached to this report (Video 1). Note that all the videos were stitched together into one long compilation for the sake of convenience.

Next, the DMM was used to record both the Base current and the Collector current at start-up and saturated speeds. These currents were recorded into **Table 2**. The saturated current may be inexact due to it being measured using the sound of the motor. On the other hand, the Start-up measurement was recorded at the current that the motor began to start rotating.

Finally, the MSP430FR2433 was used for the last part of this section. An assembly project was made to drive the motor whenever one of the built-in buttons was held. This was an accident however, as the as the intent of the project was to drive the motor whenever one of the built-in buttons was pressed, not held. This could have been easily done by setting the interrupt to toggle a flag (e.g., it would toggle a bit in R4) instead of directly toggling the output bit of the LED, and the flag would toggle the LED in the main loop. This way, the interrupt would have to end for the LED to turn on, meaning that the LED will only be on when the button has been let go. Even so, the program worked well enough, and a video was taken of it and added to the compilation (Video 2). The assembly code for this project has been attached to this report: 'Lab4_A1'.

A second coding venture was started where the program cycles through four different speeds of the DC motor using the button. The four speeds were defined by retaining the same period and shifting the duty cycle of the motor. The duty cycle and period values are as follows:

	Speed 1 (Slowest)	Speed 2	Speed 3	Speed 4 (Fastest)
Period	0.1 ms	0.1 ms	0.1 ms	0.1 ms
+ Duty Cycle	10%	30%	60%	90%

Table 1: *Period and Positive Duty Cycle of the 'Lab4_A2' code.*

Looking back, another mistake was made here as the intent was that the first state would be a stationary one. This mistake could have also been easily solved by setting the Speed 1 duty cycle to 0%, as then the motor would have remained stationary for that speed setting. A video of this system was recorded (Video 3) and the code ('Lab4_A2') was attached to this report. Please note that most of this was done at the last minute.

After this, the transistor materials were put away.

THE SERVO

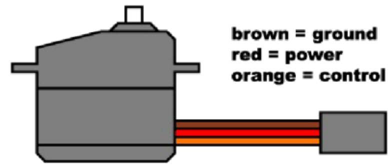


***Figure 4**: Servo Inputs*

For this part of the report, the servo was investigated. The servo utilizes Pulse Width Modulation (PWM) with a frequency of about 50 Hz and a pulse width of about 1-2 ms to control movement between 0 and 180 degrees.

Using a square wave on the function generator at a range of 0 to 3.3 Volts and with a frequency of 50 Hz, the pulse width range of two servos at the lab were measured: these servos were the robot gripper mechanisms (S06NF STD). The measurement was done by slowly moving the pulse width to one extreme until the motor hit its physical limit, and then backtracking a little bit and recording that pulse width. The same thing was done for the other extreme to record the full range of pulse widths possible for that servo. The measured pulse widths were recorded in Table 4 of the Results section.

Next step was to control the servos using the MSP430FR2433. <u>The idea for the program was to simply toggle between the minimum and maximum PWM values of the servo.</u> An assembly program was written that utilized both Clock Control Register 1 and 2 (CCR1 and CCR2) of Timer A. Using these separate Clock Control Registers, it was possible to output two separate PWM signals using the same clock. Outside of this, the code included an interrupt that was triggered by pressing a P2 button, and using the P2 Interrupt Flag Bits it was possible to distinguish which button was pressed. However, a mistake was made in this code: it had most of its logic written in the main loop when the intent was for the logic to be in the Interrupt Service Routine (ISR). This could have been done by simply moving most of the code to work under the ISR and clearing the interrupt flags at the appropriate times. Nonetheless, the code provided ('Lab4_B1') worked, and a video of the system was added to the compilation (Video 4).

Going back to the setup, the circuitry of these servos was simple. The gripper servos had three inputs: control, power, and ground (Figure 4). The gripper servos required approximately 6 Volts of power from the triple-output power supply. The ground wire was grounded, and the control wire was connected to either the P1.1 or P1.2 GPIO pin of the MSP430FR4233.

THE STEPPER MOTOR AND H-BRIDGE

The last components in this report are the Stepper Motor and H-Bridge, and these were by far the most difficult components to work with.

The motor used in this lab was a 5 Volt, Bi-Polar Stepper Motor (5V 28BYJ-48). A new assembly project was created for the purpose of controlling the stepper motor. This assembly project needed to shift between eight half-step motor states for the Bi-Polar stepper motors. In a sense, the microcontroller would have to work similarly to that of a stop light state machine, but one that is changing states incredibly fast. In order to troubleshoot and debug the code, the four outputs were first connected to a set of four LEDs. Moreover, to keep the program simple, all the outputs were placed right next to each other on Port 1. This meant that P.0 through P.3 were used for the stepper motor outputs of the MSP430FR2433.

The next large hurdle was utilizing the H-Bridge. Using the diagram from the Laboratory Lecture notes (Figure 5) and with some trial and error, the motor began to rotate. The key factor was using the proper order of connection between the MSP430FR2433 GPIO and the Stepper Motor at the H-Bridge. If even one wire was not connected properly, then the stepper motor would not rotate, but vibrate. Once the motor began rotating, a video of it was taken and added to the compilation (Video 5).

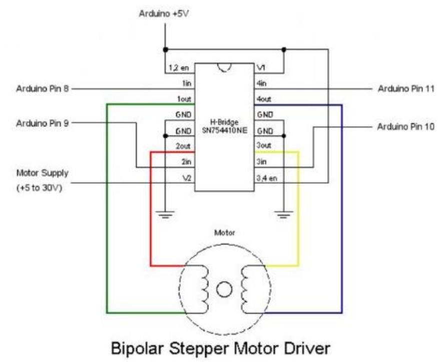


***Figure 5**: Stepper Motor Circuit Diagram from Lecture*

After this, the objective was to adjust the program so that it would toggle between the direction of the motor's rotation and change the speed at which the motor rotated at.

To toggle the direction, a second state table was made that was simply an invert of the first one, and a button ISR would toggle a flag that would decide which state table would be used to rotate the motor. A video of this system was also added to the compilation (Video 6).

To adjust the speed, a second option was added to the ISR for a separate button flag. For this button, the ISR would adjust the maximum value of the Timer A register: the shorter the maximum value, the faster the motor would rotate. A video of this system was taken and added last to the compilation (Video 7). The code for this system has also been attached to this report: 'Lab4_B2'.

With this, the project was over, and all the supplies were put back in place.

RESULTS AND DATA

THE TRANSISTOR

V_{IN}	V_{CC}	V_O (V_{CE})	V_B (V_{BE})	R_B (10k)	R_C (1k)	I_B	I_C	beta (I_C/I_B)
0	11.99	11.98400	0.000	9864	985	-3.04136E-08	6.09137E-06	-200.284264
0.2	11.99	11.98400	0.200	9864	985	0.00000E+00	6.09137E-06	#DIV/0!
0.4	11.99	11.98300	0.400	9864	985	0.00000E+00	7.10660E-06	#DIV/0!
0.6	11.99	11.29500	0.556	9864	985	4.46067E-06	7.05584E-04	158.1790494
0.8	11.99	8.39500	0.595	9864	985	2.07826E-05	3.64975E-03	175.6151046
1	11.99	5.09400	0.611	9864	985	3.94363E-05	7.00102E-03	177.5270288
1.2	11.99	1.81700	0.623	9864	985	5.84955E-05	1.03279E-02	176.559083
1.4	11.99	0.11930	0.631	9864	985	7.79603E-05	1.20515E-02	154.5848122
1.6	11.99	0.08701	0.634	9864	985	9.79319E-05	1.20843E-02	123.3944923
1.8	11.99	0.07280	0.636	9864	985	1.18005E-04	1.20987E-02	102.5269601
2	11.99	0.06363	0.636	9864	985	1.38281E-04	1.21080E-02	87.56100576
3	11.99	0.04102	0.642	9864	985	2.39051E-04	1.21309E-02	50.7462397
4	11.99	0.03091	0.647	9864	985	3.39923E-04	1.21412E-02	35.71752965
5	11.99	0.02501	0.651	9864	985	4.40896E-04	1.21472E-02	27.55115217
6	11.99	0.02113	0.655	9864	985	5.41869E-04	1.21511E-02	22.42447445

***Table 2**: Data collection sheet for the ZTX651 NPN transistor. Includes V_{IN} : Input Voltage; V_{BE}: Base-Emitter Voltage; I_C: Collector Current; I_B: Base Current; and V_O: Output voltage.*

In the lab manual, a question was posed: <u>Is the beta value the same at different values of the collector current (I_C)?</u> Our data table shows that <u>this is not the case</u>. As the collector current decreases, the value for beta also decreases. This makes sense as beta is directly related to the collector's current value. Moreover, from this table, <u>the average beta value was calculated to be around **107.7**.</u>

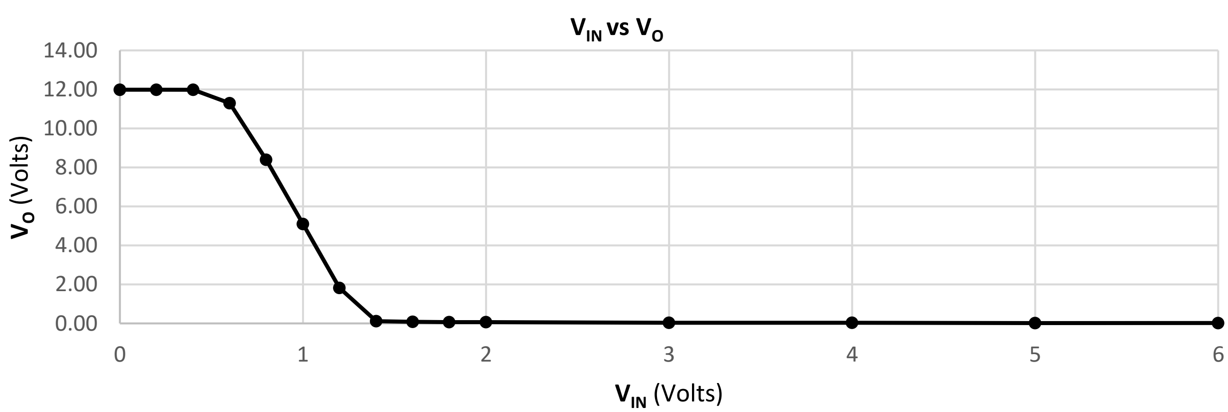


***Figure 7**: The change in the Collector-Emitter Voltage (V_o) as the Input Voltage (V_{IN}) increases.*

<u>This graph shows that our saturation voltage $V_{CE(sat)}$ is at approximately 1.2 Volts of input voltage, and our active voltage starts at 0.5 Volts of input voltage.</u>

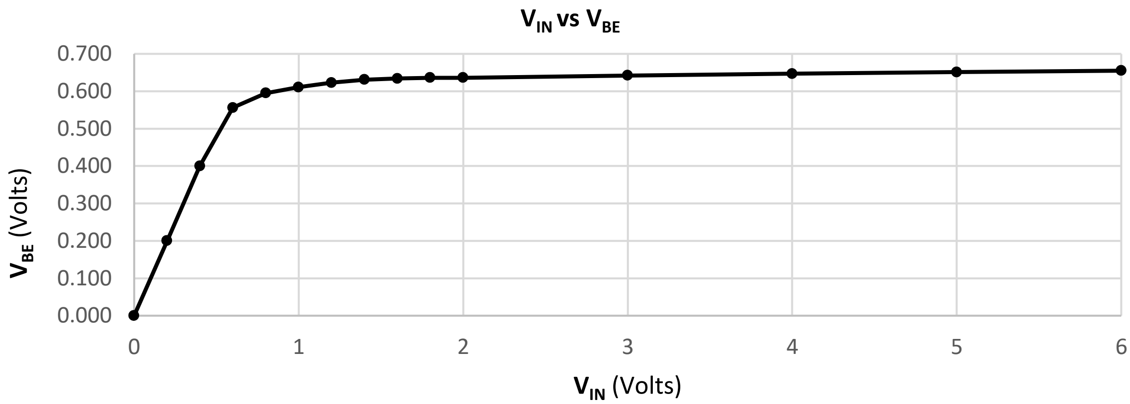


***Figure 8**: The change of Base-Emitter Voltage (V_{BE}) as the Input Voltage increases.*

Another question was posed in the lab**:** Approximately what is the voltage V_{BE} when the transistor is in saturation? What practical limitations, if any, does this place on the voltage used to drive the common emitter circuit when the transistor is to be used as an amplifier? As a switch?

At 1.2 Volts, when the Collector-Emitter (V_O) voltage is at saturation, the Base-Emitter voltage seems to be at about 0.65 Volts. For **amplification**, it would be good to minimize the V_{BE} to reduce distortion, and it would be necessary to be in the active region of the transistor. Hence, one should consider using the 0.5 to 1.2 Volt input range. However, in the case of a **switch**, it is good to reduce the V_O as much as possible to reduce energy consumption, hence an input Voltage of 1.2 or above would be better.

	Collector	Base
Start-up Current	84 mA	0.4 mA
Saturated Current	120 mA	1.2 mA

***Table 3**: Base and Collector currents for startup and saturated motor speeds.*

THE SERVO

Gripper Servo	Max. Pulse Width	Min. Pulse Width
Arm of Gripper	2.4 ms (12%)	0.4 ms (2%)
Hand of Gripper	2.4 ms (12%)	0.5 ms (2.5%)

***Table 4**: The maximum and minimum Pulse Widths for the gripper servo.*

DISCUSSION AND CONCLUSION

In this lab report, the fundamental principles and practical applications of interfacing with the world using the MSP430FR2344 microcontroller were explored. Experiments with the transistor showed its ability to function as an electrical switch. Additionally, investigation into servo motors demonstrated the utilization of Pulse Width Modulation (PWM) to control movement within a specified range. Moreover, the challenges encountered and successfully navigated with the stepper motor and H-Bridge assembly displayed the complexity involved in controlling bi-polar stepper motors.

In conclusion, this lab not only provided valuable hands-on experience with essential electronic components but also honed critical problem-solving skills necessary for effective embedded systems design. Ultimately, the comprehensive study of interfacing techniques gave insights into real-world applications, laying a solid foundation for future projects in embedded systems development.