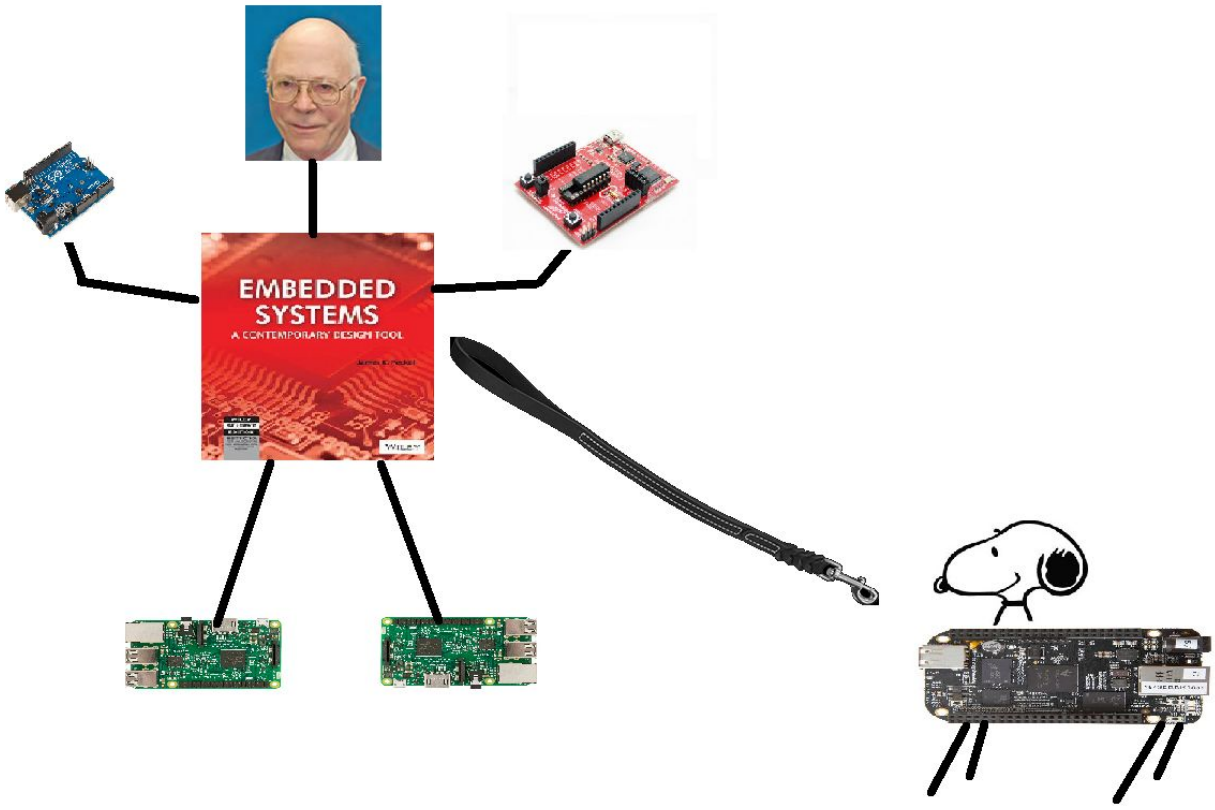
EE 474: Lab Report 4

Michael Walsh, Josh Shackelford, Alex Finestead

TABLE OF CONTENTS



Introduction:

In this fourth lab assignment, we used a collection of sensors to control the movement of the motors on our robo-tank. Analog readings were taken from four distance sensors mounted on the tank chassis which were then interpreted and used to control the flow of our program. The motors of the tank were governed by an H-bridge chip mounted on the breadboard atop the tank, and powered by a set of AA batteries. The main tank process includes driving forward when it senses something above it, stopping when it sees a wall or another object in its path, and then waiting for another object to appear above it before driving back in the opposite direction.

Design Specifications:

The design specifications were intentionally vague for this particular assignment. We were asked to use the optical sensors in our kit as a way of receiving analog input to the BeagleBone, as well as to use the provided H-bridge chip to drive the motors of the tank. We chose to use the readings from the sensors as a way of driving the tank in a semi-autonomous fashion.

Hardware Implementation:

The BeagleBone Black had to be wired up to take four distinct analog inputs and control the H-bridge. Firstly the optical sensors were installed and wired. Voltage dividers were built for each sensor as a way of reducing the output voltage of the sensor. ### kΩ resistors were used to drop the maximum output voltage of each sensor to about 1.4V. This reduced voltage was seen to give us significantly more accurate than when the sensors were left at an unimpeded 1.8V. The data lines for each of the sensors were wired into the analog to digital converter pins on the BeagleBone as described in **Table 1**.

With the sensors in place, the motors were the next piece of hardware added to the design. The H-bridge chip was placed onto the breadboard, and its pins wired to up as shown in **Figure 1**. The connections of the H-bridge to the BeagleBone can also be found in **Table 1**, below.

BeagleBone Black Pin	*Function*
AIN0	Back Sensor
AIN2	Front Sensor
AIN4	Top Sensor
AIN6	Bottom Sensor
P9_14 (1A)	PWMA
P8_19 (2A)	PWMB

GPIO_60	AIN1
GPIO_48	AIN2
GPIO_115	BIN1
GPIO_49	BIN2
GPIO_112	STDBY

Table 1: BeagleBone Black Pin Assignments

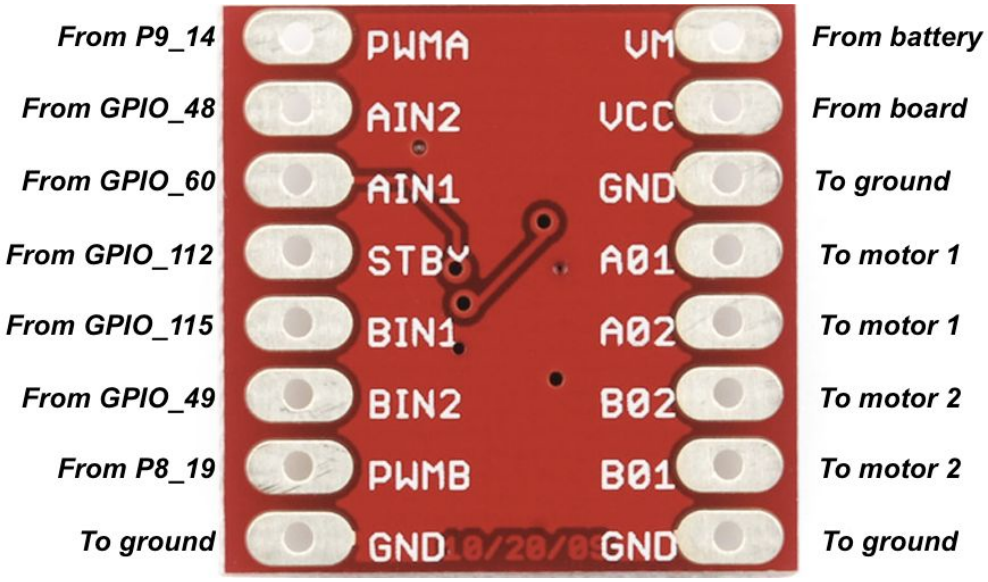


Figure 1: H-bridge connection diagram

IN2	IN1	Behavior
0	0	“Free”, Motor is high z and does not hold the wheel
0	1	“Forward”, The motor drives the wheel in a “forward” direction
1	0	“Reverse”, The motor drives the wheel in a “backwards” direction
1	1	“Brake”, the motor holds the wheel, acting as a brake

Table 2: Truth Table for H-Bridge Motor Control (Same for A and B)

This was the extent of the hardware engineering for this lab project. There were no software initializations that were required to run the devices, apart from the standard requesting of GPIO pins, PWM, and ADC initialization.

Software Implementation:

One of the main components of this project was to implement a software interrupt as a way of controlling the program. A system timer was created and used as our interrupt for this lab. The signal handler was called on every expiration of the timer, which then polled the sensors and changed the movement of the motors accordingly. The state diagram of the robot logic can be seen in **Figure 2**.

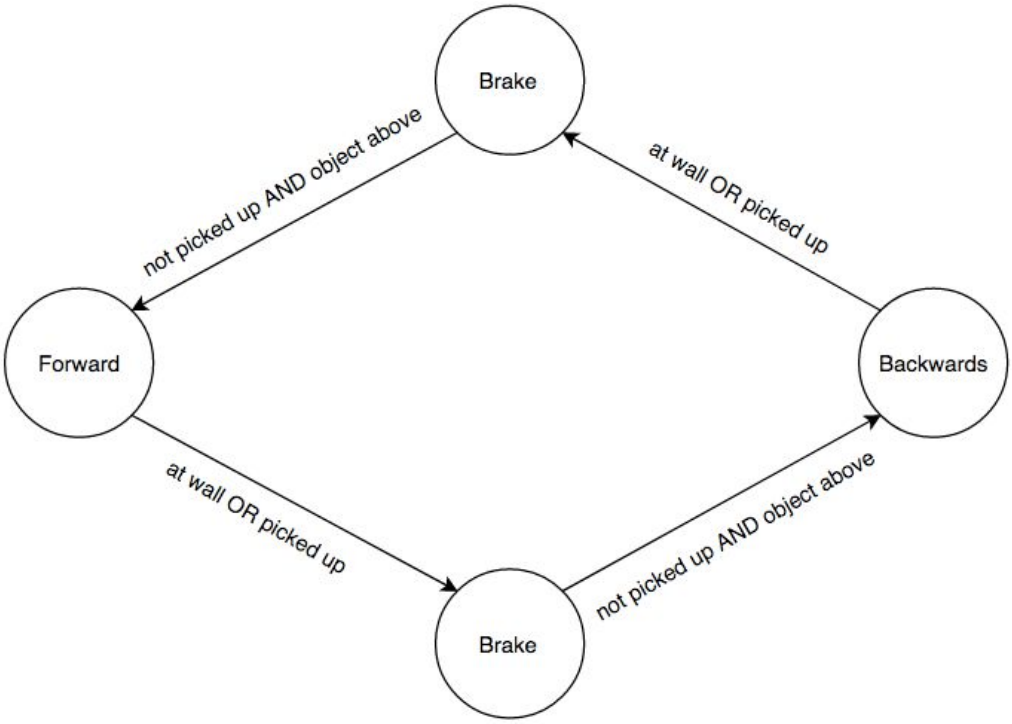


Figure 2: Motor State Machine

In plain language, the tank waits until it senses an object above it. It then moves forward quickly until it senses an object about 10cm in front of it. It then waits until it senses another object above it, and drives in the opposite direction until there is something 10cm in front of it. The process repeats indefinitely. If the robot ever gets picked up while it is driving, it stops moving its tracks and waits to be set down again before continuing.

Polling of the sensors was performed inside the signal handler. Whenever the program received the signal to check its sensors, the values on the ADC pins were read and sent into the state logic.

To better interact with the motors, a library was written on top of the main process. This library included initialization processes of the GPIO pins and PWM's, functions for driving forward, backwards, spinning left and right, and braking.

There is no user space program for this project; the tank is fully autonomous. In addition, this program runs on bootup of the BeagleBone Black, abstracting all programming responsibilities away from the user.

Testing:

All testing was performed in the lab. As each sensor was connected, an oscilloscope was attached to the outputs to ensure that the sensor was reacting properly to objects in its field of view and ensure that the output voltage would not exceed the 1.8V limit on the Beaglebone's ADC inputs. We determined that the sensors were most accurate when given a power supply of 5V; however, at this voltage they output up to 3V. Thus we built 1:1 voltage dividers that would cut the output of the sensors in half; meaning the ADC inputs would not be damaged by the sensor readings.

With the sensors fully linked into the board, we wrote a program that printed out each of the four (4) sensor values to the console every tenth of a second. This was used to ensure that the sensors were constantly updating and reading the environment correctly, as well as testing out timer-based interrupt code.

As the motor control program was being written, each iteration was tested to see if it moved the motors (and therefore the tires) in the desired way. The tank was set up on a block so that the wheels could spin freely without moving the tank, and all functionalities were tested. Once the sensors were connected and implemented to change the motor behavior, the system was then tested by setting it up on blocks and allowing it to run. We brought notebooks and hands close to different sensors to ensure that the tank system behaved as we had designed.

Presentation of Results:

The final product for this lab consisted of a motorized tank with four sensors attached: one on the front, one on the back, one on the bottom, and one on the top. The front and back sensors ensured that the tank would not crash into objects in its line of motion. The bottom sensor was meant to stop the motors when the tank was picked up. The top sensor behaved as a start signal, when an object approached from above, the tank would begin moving forward until it was about to hit an object. The next time an object was detected above, the tank moved in the opposite direction. See **Table 3** below for a description of sensor operations.

Front Sensor	Stopped forward motion when an object got within ~3 inches
Back Sensor	Stopped reverse motion when an object got within ~3 inches
Bottom Sensor	Stopped motors when an object cannot be detected within ~3 inches

Top Sensor	Begins motor control when an object is detected within ~2 inches

Table 3: Description of Sensor Operations

Analysis of Errors:

The majority of errors for this lab were hardware related. One of the biggest hurdles we had to overcome was the inaccuracy of the optical sensor readings. We noticed that reading would vary widely for the same sensor, even under controlled conditions. We approached this particular issue by inserting voltage dividers for each sensor. This lowered the output voltage of the sensors (the input voltage to the BeagleBone) to around 1.4V. This reduction in voltage gave us much more accurate sensor readings.

Another error that was realized in this lab was that the tank would, on occasion, move very slow, even when we were setting its motors to maximum speed with fully charged batteries. We are still unsure on the source of this error, but found that by resetting the board, this error was temporarily resolved. A possible cause is a memory leak, which would eventually slow down the processor. Further testing with an valgrind or similar software could pinpoint the error.

The Beaglebone would occasionally shut off and would not reboot. After testing, this was shown to be a high voltage being driven to the analog inputs. With bare wire exposed, sometimes five volts was exposed to the ADCs, and the board was shut off. This was resolved by wrapping tape around wires exposed near to each other.

Conclusion:

In this fourth lab, the instructions became even more open-ended as we gain more familiarity with our software and hardware systems. We were instructed to use 4 Analog to Digital Converters as well as an H-Bridge to drive the two wheel-motors, but otherwise had full freedom of design. The labs are meant to test our designing abilities, both in concept and implementation. The system we designed used sensors to monitor the outside world and provide digital information that we could react to.

With the completion of this lab, we have further improved our ability to work with the Beaglebone and the hardware kit we have been given. This lab introduces a variety of interesting and cool possibilities for our final project which is even more open-ended than this past one.