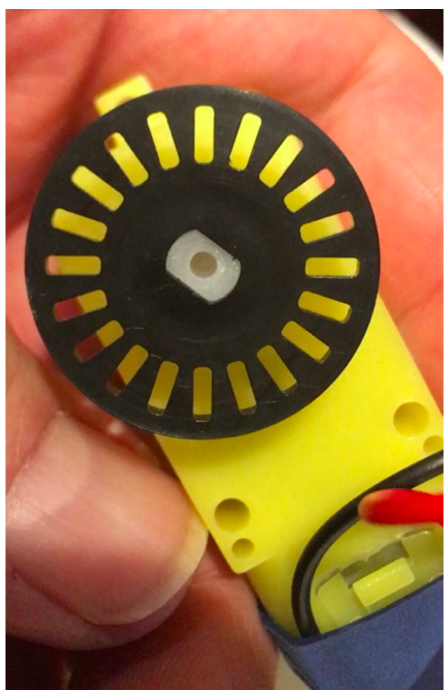
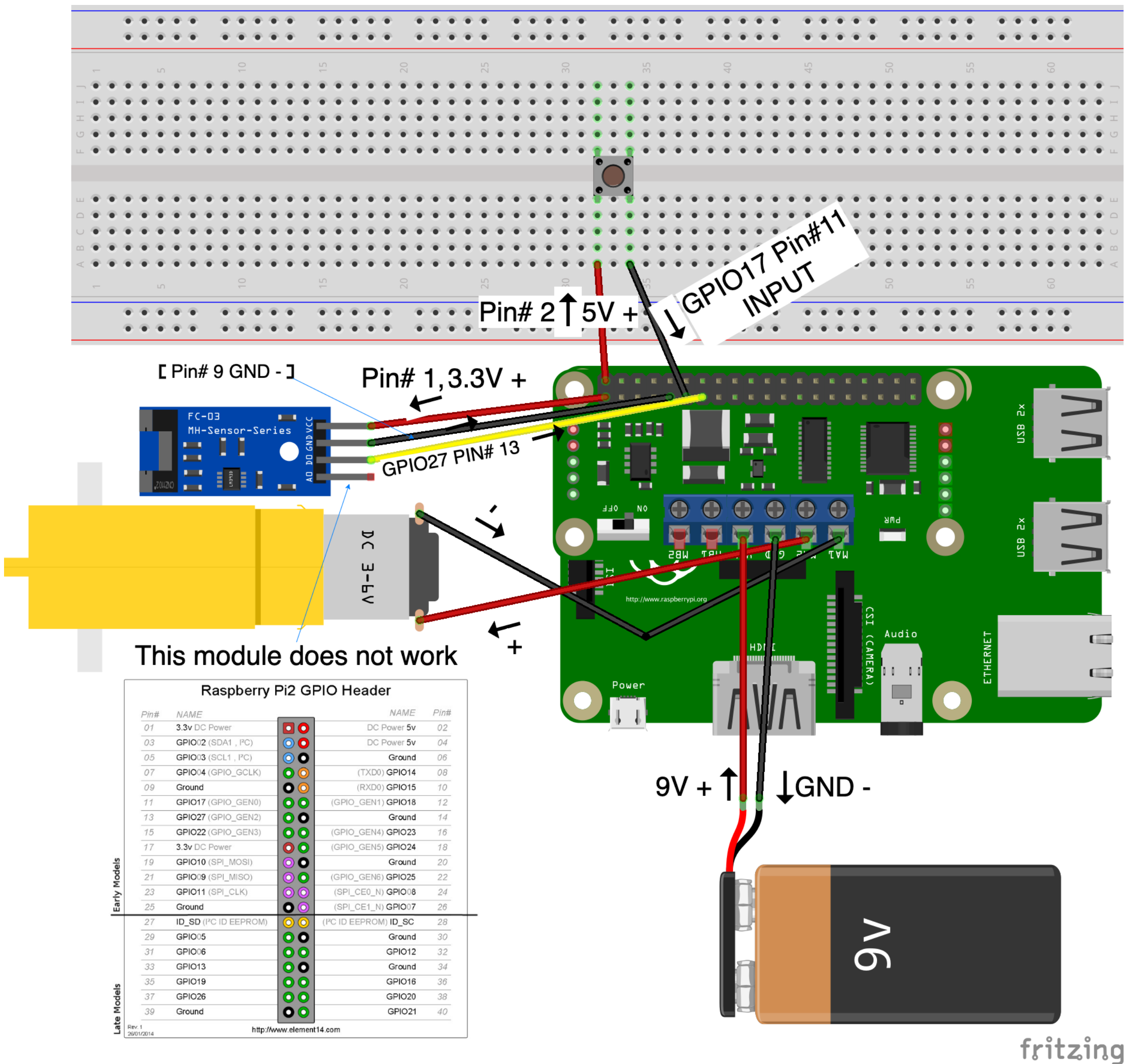
CSC 615 Assignment 4 – Motors & Speed Encoder

Wameedh Mohammed Ali
ID: 920678405
GitHub ID: wameedh

Note: I couldn't find an encoder wheel pic for the hardware diagram but you would see how it installed it in the demo video and also as shown in the picture below for:



Hardware Diagram:

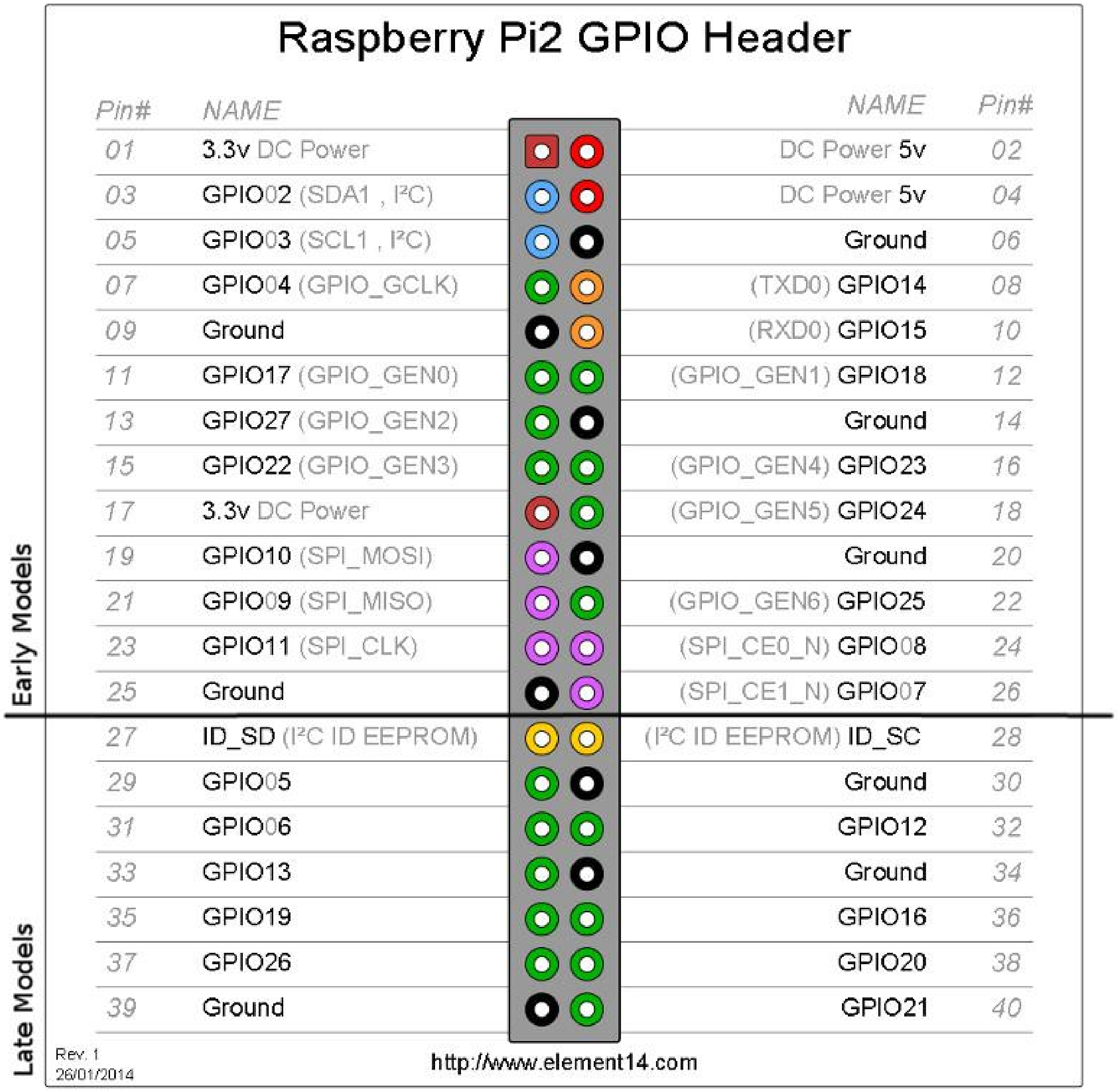


Raspberry Pi2 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Early Models (pins 01–26) / Late Models (pins 27–40)

Rev. 1
26/01/2014
http://www.element14.com



How it works:

The speed behavior of the motor is the same from assignment 3, which is as fellow:

> Using the I2C interface with the Waveshare motor controller Hat board this program controls the motor. Using a button to start the motor the program would listen to a signal that the button has been pressed, then it would run the mortar

forward for about 5 seconds. It would then slow down gradually to 15% , then stop the motor for about one second, then the motor would start again but this time backward. The speed would gradually increase to max.

As the motor speeding up and down and down the program runs a thread that monitor the data coming from the speed sensor, calculating the angular and the linear speed using the following formulas:

$$\omega = 2\pi f/Nm$$

Where:
ω = angular speed (rad/s)
f = clock frequency (Hz)
m = number of clock cycles
N = pulses per rotation

$$speed = rw$$

Where:

R = radios of the wheel.

W = angular speed.

Hardware used:

1. Raspberry Pi 4 model B
2. WaveShare Motor Drive HAT
3. 2 male-to-female jumper cable
4. 3 female-to-female jumper cable
5. 9V battery
6. 9V battery wire connector
7. Breadboard
8. TT DC Gearbox Motor
9. Speed sensor
10. Encoder wheel

Softwear:

The software written in C. There is one version of the code, it uses the WiringPi library (http://wiringpi.com/.) It also uses the math library and the pthread library.

How to run the program:

Follow the diagram provided for setting up the hardware connections. Clone the repo from github to get the program on the Raspberry Pi,
https://github.com/CSC615-Spring2021/assignment-4-motors-and-speed-encoders-Wameedh.git
After cloning the repo cd to the directory to run the code.
Use the command `make run` to run the program, then the program would prompt in the console the following message, “Press Button to start…” then it would wait for the user to press the button to start.

Challenges:

The assignment was challenging to setup at first because I thought that a chip is needed to be part of the hardware setup. The documentation provided for the assignment unfortunately is not clear in terms of how the individual version of the assignment looks like vs the hardware manager version. I was able to figure out how to setup the project by reaching out to the professor who was very helpful and responsive.

Another big challenge was the timing of the pulse. I noticed that the runSpeedSonser() method depends on the value of “the time to measure”. I kept getting discrepancies between the power that was being applied and the speed produced. I was able to get to a good enough speed calculator by applying trial and error technique.