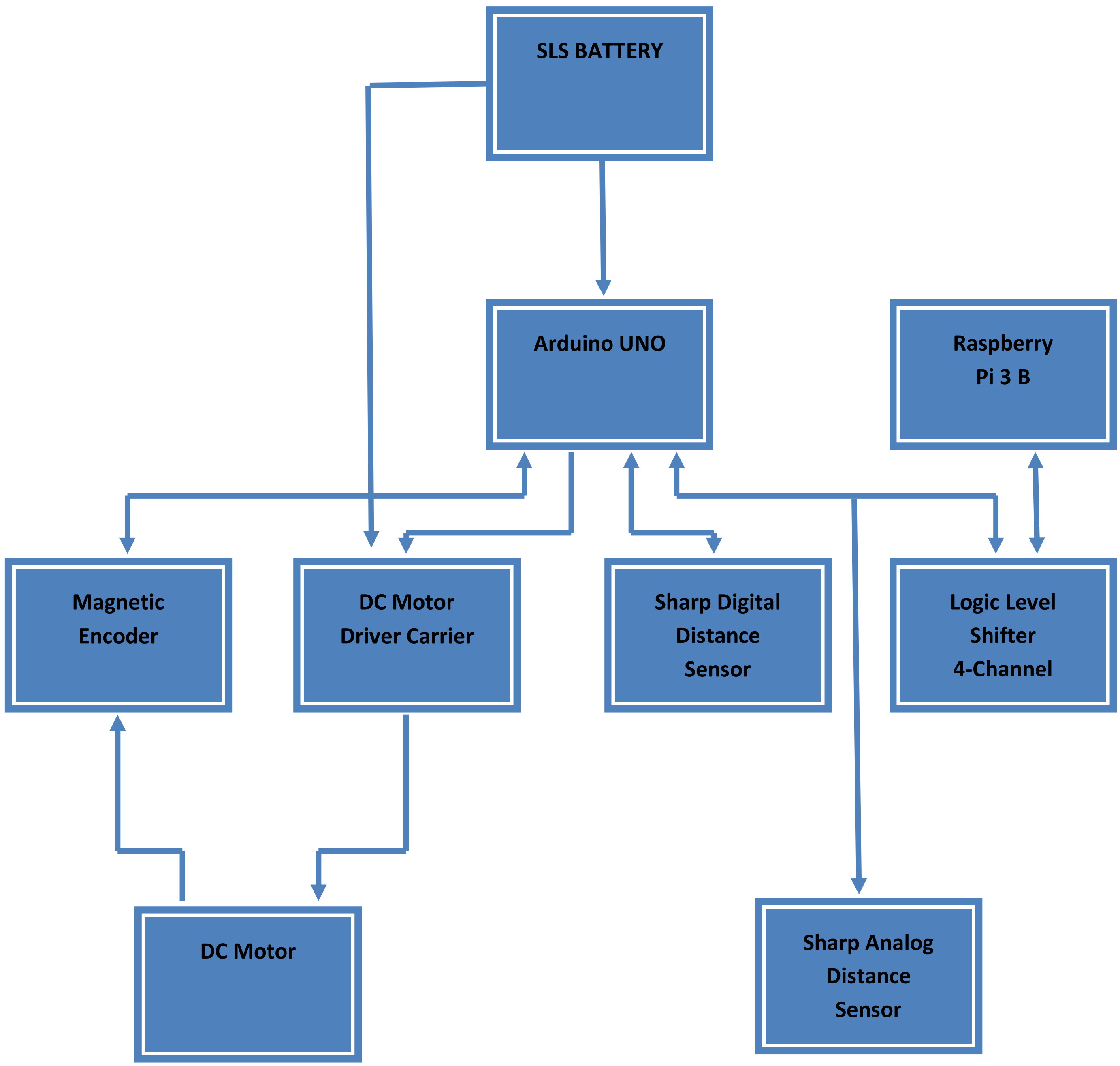
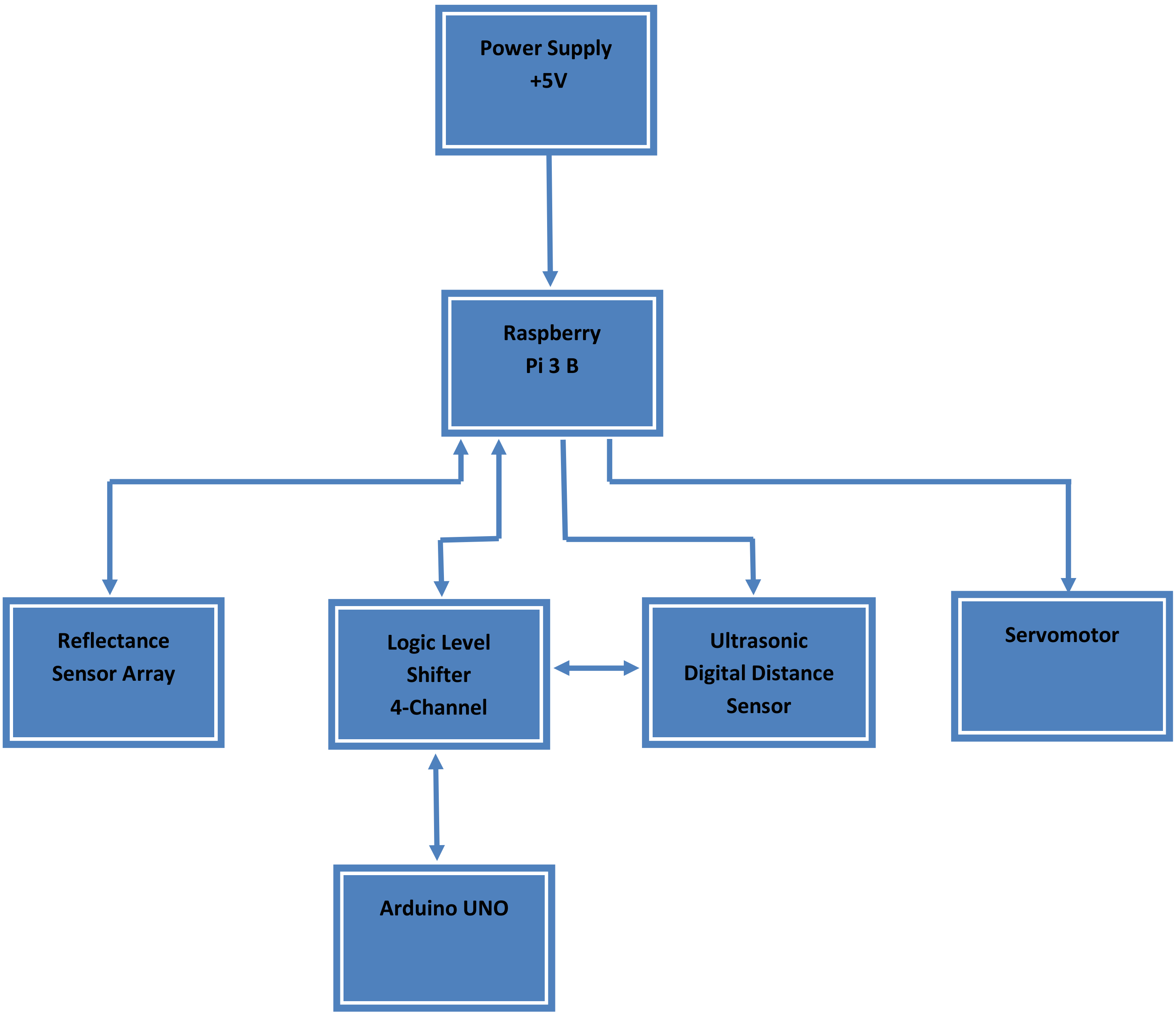
SYSTEM ARCHITECTURE CONNECTIVITY ARDUINO UNO

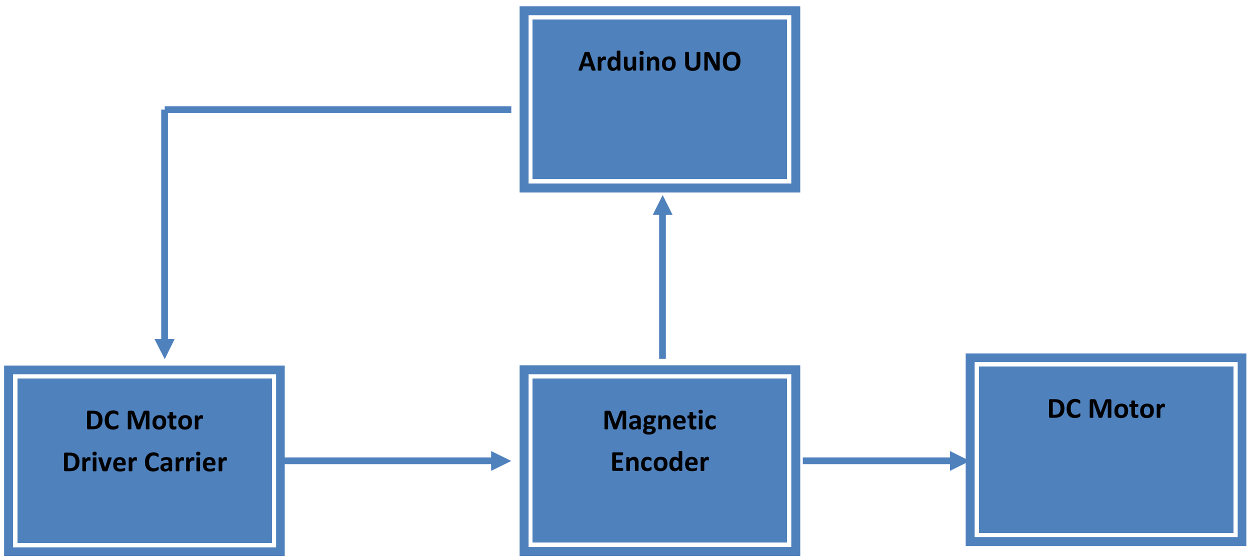


SYSTEM ARCHITECTURE CONNECTIVITY RASPBERRY PI 3 B

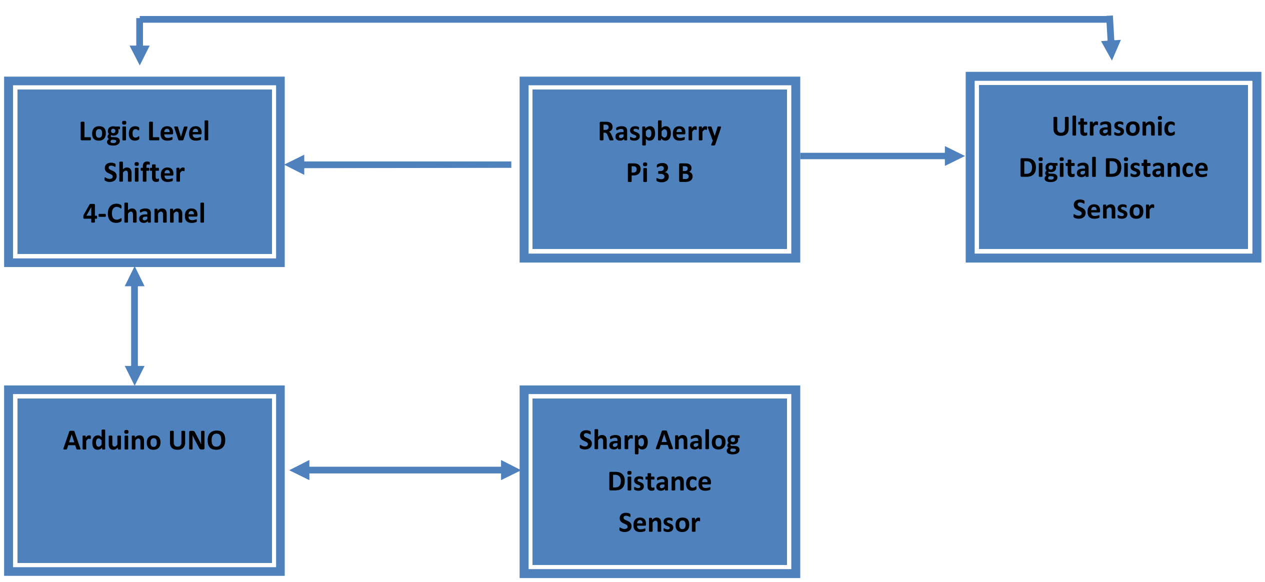


BLOCK DIAGRAM

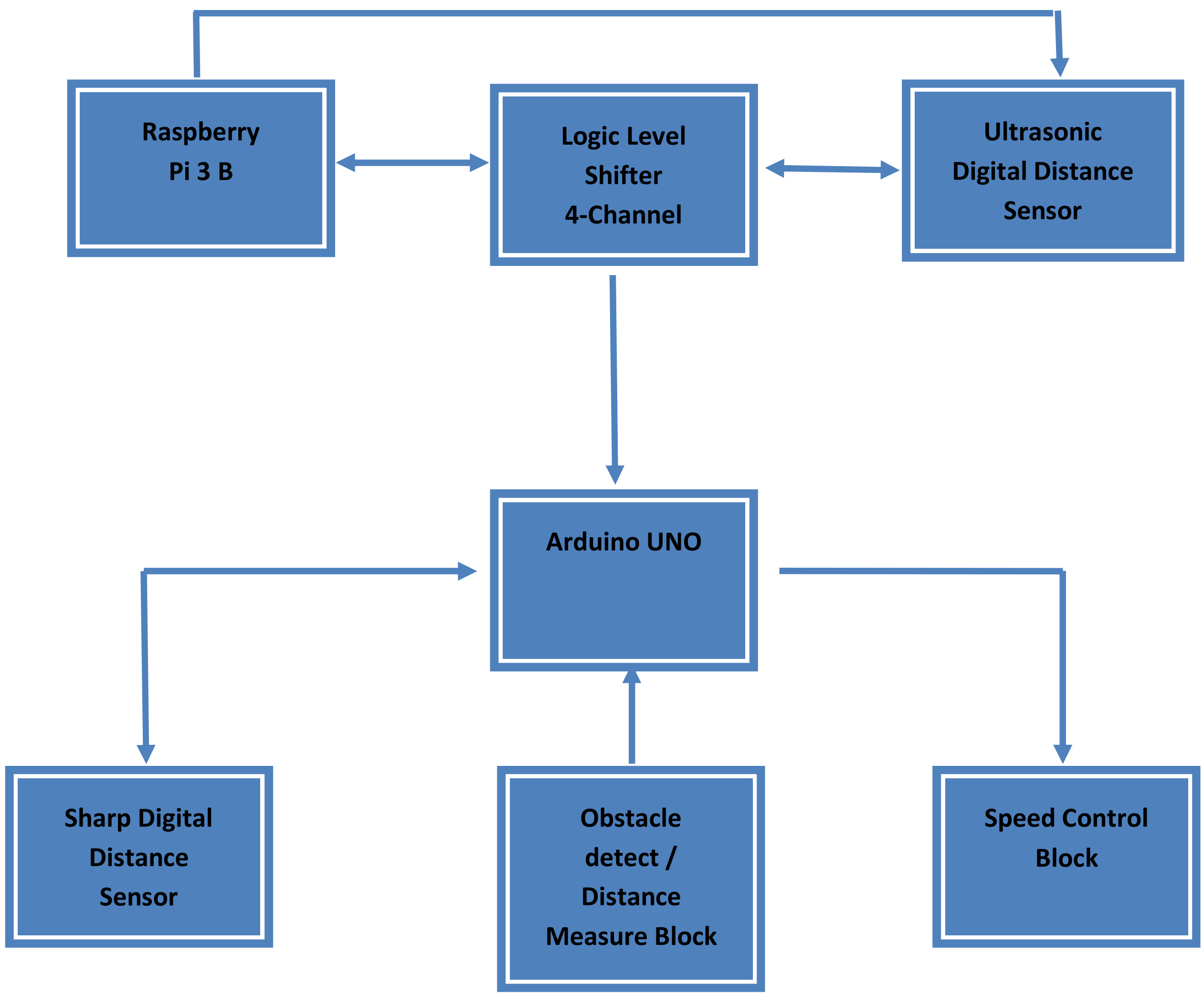
1. Speed control block:



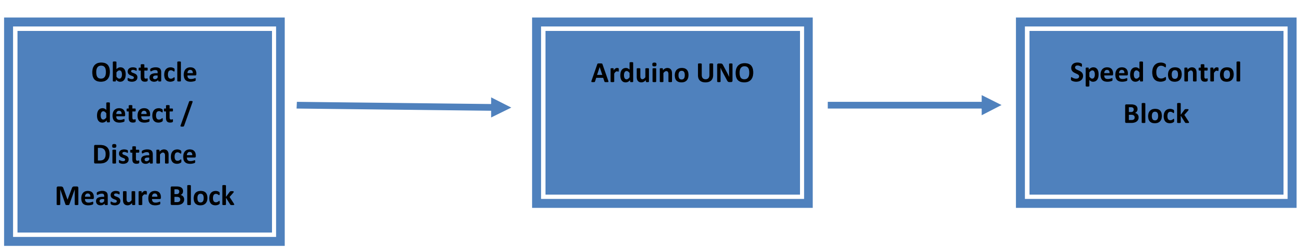
2. Obstacle detect / Distance measure block



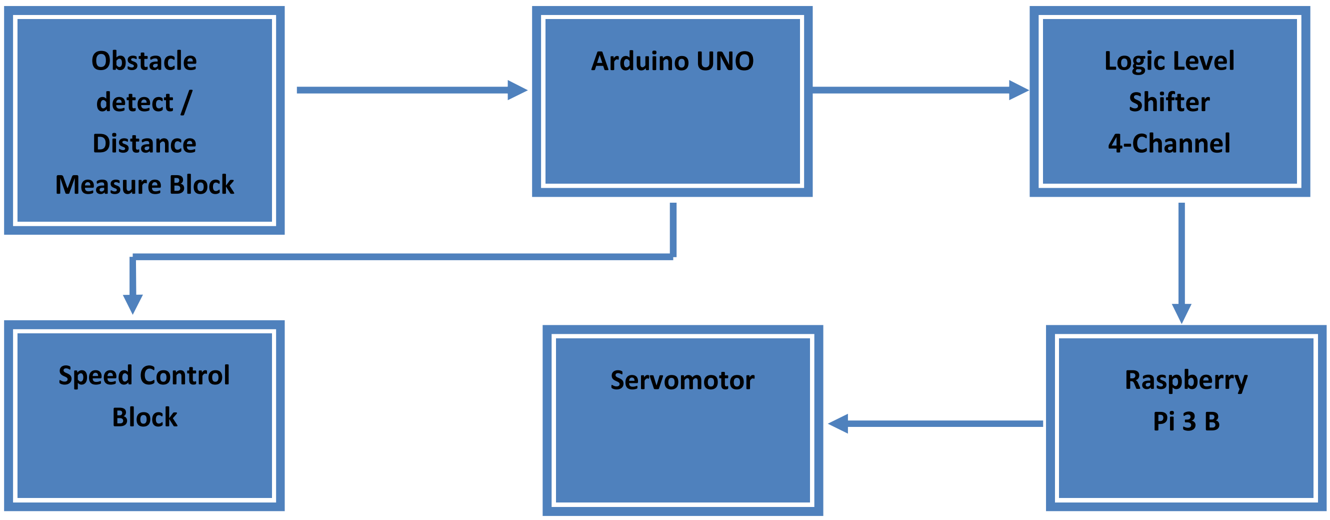
3. Following target block:



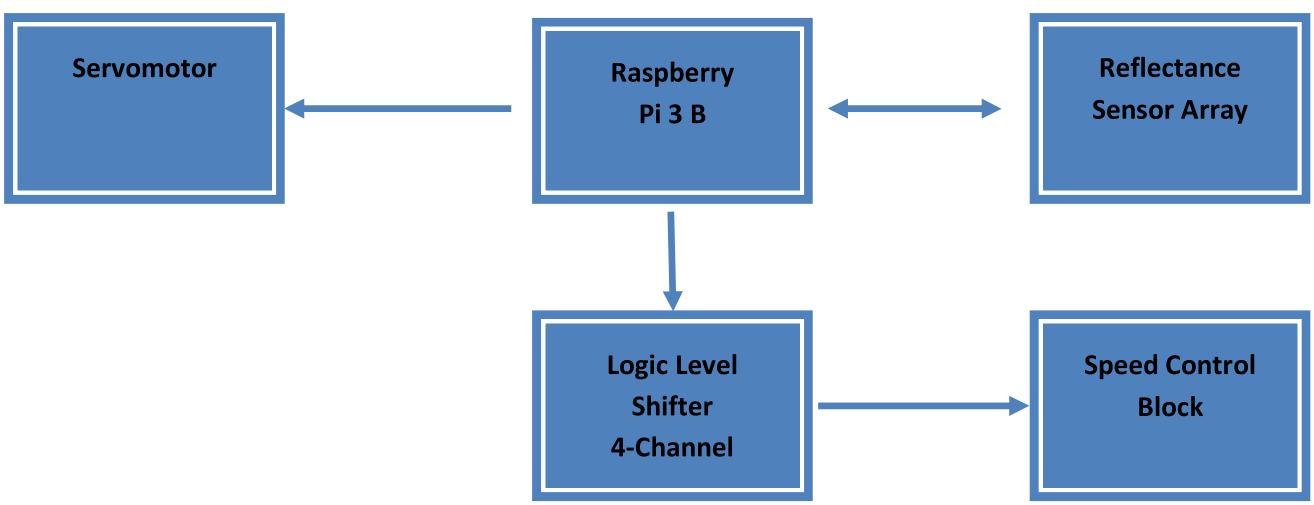
4. Stop obstacle block:



5. Avoidance obstacle block:



6. Following line block:



INPUT OUTPUT DESCRIPTION

DC Motor: It has 2 pins, a power supply (VIN) and a ground (GND), which are connected to the encoder M1 and M2 holes.

Encoder: It has 6 pins, 4 of them are connected to the DC Motor Driver, the power supply (VCC), the ground (GND) and the M1 and M2 pins that feed the motor and give the direction of rotation. VCC can be 2.7V to 18V, and quadrature outputs A and B are digital signals that are either driven low by the sensors or pulled to VCC through 10kΩ pull-up resistors, depending on the applied magnetic field. The two output signals are displaced at 90 degrees out of phase from each other. If the output A is in front of the output B, the motor is rotating clockwise and if the output B is forward, the motor is rotating counter clockwise.

DC Motor Driver: It has 10 pins, of which only 7 are used. The M1 and M2 pin connected to the encoder, the PWM and DIR pins connect to the Arduino (pin D3 to supply the PWM signal, respectively D4 to change the direction of rotation), the VIN pin connected to VCC pin from SLS Battery. One GND is connected to SLS Battery's ground and one to Arduino's ground.

Digital Distance Sensor: It has 3 pins, all connected to Arduino. VIN connected to the 5V supply pin, GND connected to the GND pin on Arduino and OUT connected to the D7 pin. It is able to detect the presence of a barrier at a maximum distance of 10cm.

Logic Level Shifter: It has 10 pins, one (HV) for 5V supply from Arduino (5V pin), one (LV) for 3.3V supply from Raspberry Pi (3.3V pin). The other 8 pins represent conversion channels from 5V to 3.3V (HV1-HV4) respectively from 3.3V to 5V (LV1-LV4). It is used for connection between Arduino and Raspberry Pi, HV1, HV2 pins connected to Arduino SDA and SCL pins and LV1, LV2 pins connected to GPIO8 (SDA) and GPIO9 (SCL) pins. This makes the I2C connection between the two devices. Channels 3 and 4 (HV3-LV3, HV4-LV4) are used for communication between ECHO and TRIG pins of the ultrasonic sensor with GPIO24 and GPIO25 pins.

Analog Distance Sensor: It has 4 pins, VCC is connected to 3V VCC pin, GND is connected to Arduino's GND1 pin. The OUT pin is connected to A3 pin from Arduino and EN (ENABLE) is disconnected.

Digital Ultrasonic Sensor: it has 4 pins, the VCC is connected to the 5V pin, the GND is connected to one of the GND pins, the TRIG is connected to the GPIO25 via the 4th channel of the Logic level shifter, and the ECHO is connected to the GPIO24 via channel 3 the Logic level shifter, all of Raspberry Pi. The TRIG pin transmits a signal that returns to an object meeting and is received by the ECHO pin. The calculation of the distance is based on the moment when the signal was transmitted by TRIG and the moment when the signal was received by ECHO.

Servomotor: It has 3 pins, VCC is connected to 5V, GND is connected to one of the GND pins, and CMD is connected to Raspberry Pi GPIO26 pin (PWM signal pin). It is able to rotate 60 degrees in 0.11 seconds. The servomotor works with standard RC servo pulses, offering position control over an approximate 120 ° operating angle for servo pulses between 900 µs to 2100 µs.

QTR-8RC Reflection Sensor Array: It has 10 pin connected. The VCC is connected to the 3.3V power supply pin, the GND is connected to one of the GND pins, and the eight pin of the infrared sensors are connected to GPIO0, GPIO2, GPIO3, GPIO15, GPIO16, GPIO1, GPIO4, GPIO5.

Arduino Uno: It has 14 digital pins and 6 analog pins:

-D3, D4 are connected to PWM and DIR pins from DC Motor Driver.

-D5, D6 are connected to A and B pins from Encoder.

-D7 is connected to OUT pin from Digital Distance Sensor.

-A2 is connected to VCC from SLS Battery for charge level monitoring.

-A3 is connected to OUT pin from Analog Distance Sensor.

VCC, VIN, SDA, SCL and GND:

-5V VCC is used to power supply the Digital Distance Sensor and the Logic Level Shifter.

-3.3V is used to power supply the Analog Distance Sensor.

-VIN is used to power supply the Arduino Board.

-SDA and SCL are used for I2C connection with Raspberry Pi.

-GND is connected to DC Motor Driver and Digital Distance Sensor grounds.

-GND1 is connected to SLS Battery's ground.

-GND2 is connected to Raspberry Pi and Analog Distance Sensor grounds.

Raspberry Pi 3 B:

-GPIO 0, 2, 3, 15, 16, 1, 4, 5 pins are connected to the Array Reflectance Sensor outputs.

-GPIO 24, 25 pins are connected to ECHO and TRIG pins from Ultrasonic Sensor.

-GPIO 26 pin is connected to CMD pin from Servo Motor.

VCC, GND, SDA and SCL:

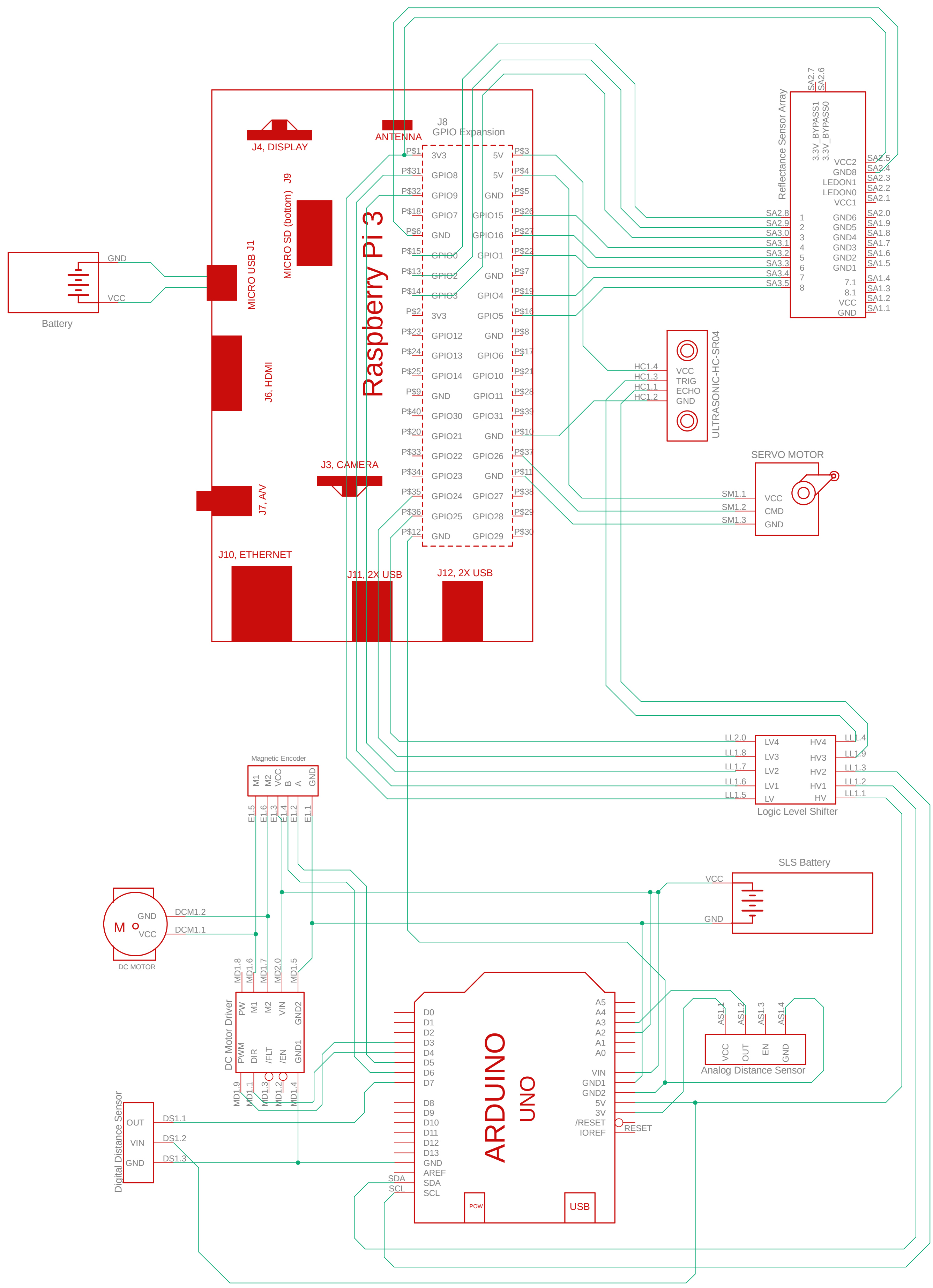
-3.3V is used to power supply the Array Reflectance Sensor and the Logic Level Shifter.

-5V pins are used to power supply the Ultrasonic Sensor and the Servo Motor.

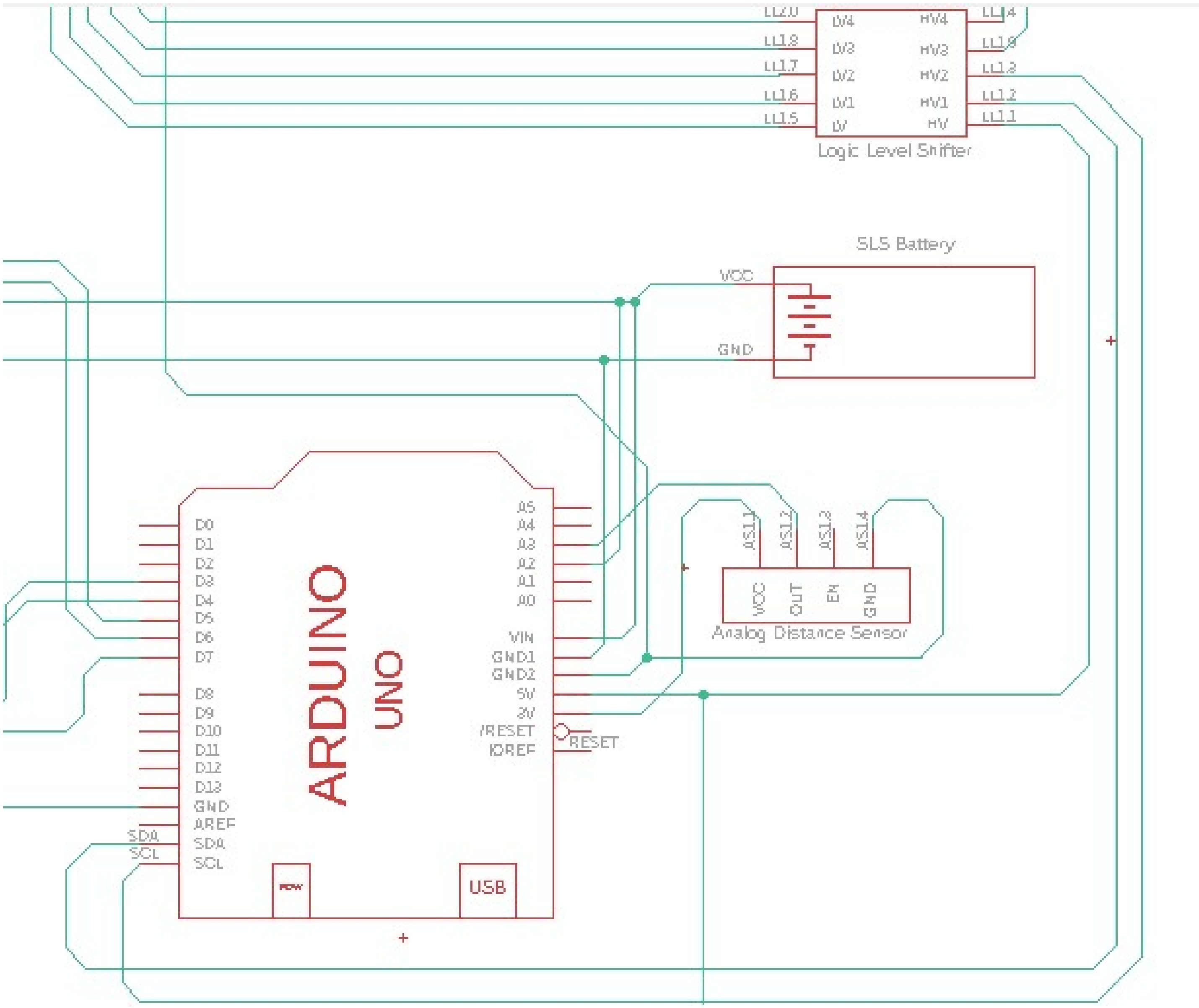
-GND pins are connected to Array Reflectance Sensor, Ultrasonic Sensor, Servo Motor and Arduino UNO.

-GPIO 8, 9 are connected to SDA and SCL pins through Logic Level Shifter 3 and 4 channels.

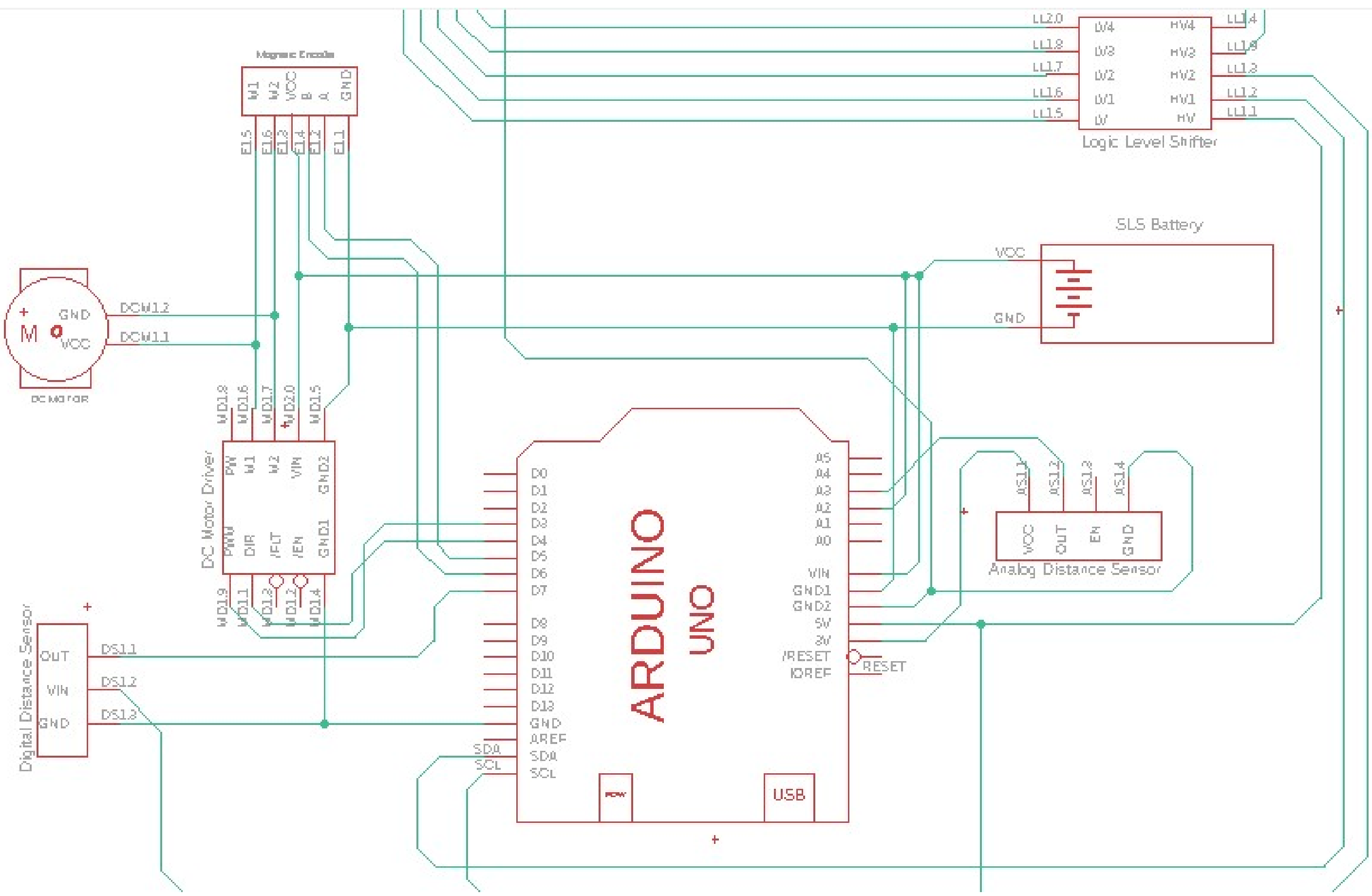
-The Micro USB port is used to power supply the Raspberry Pi from the 5V battery.

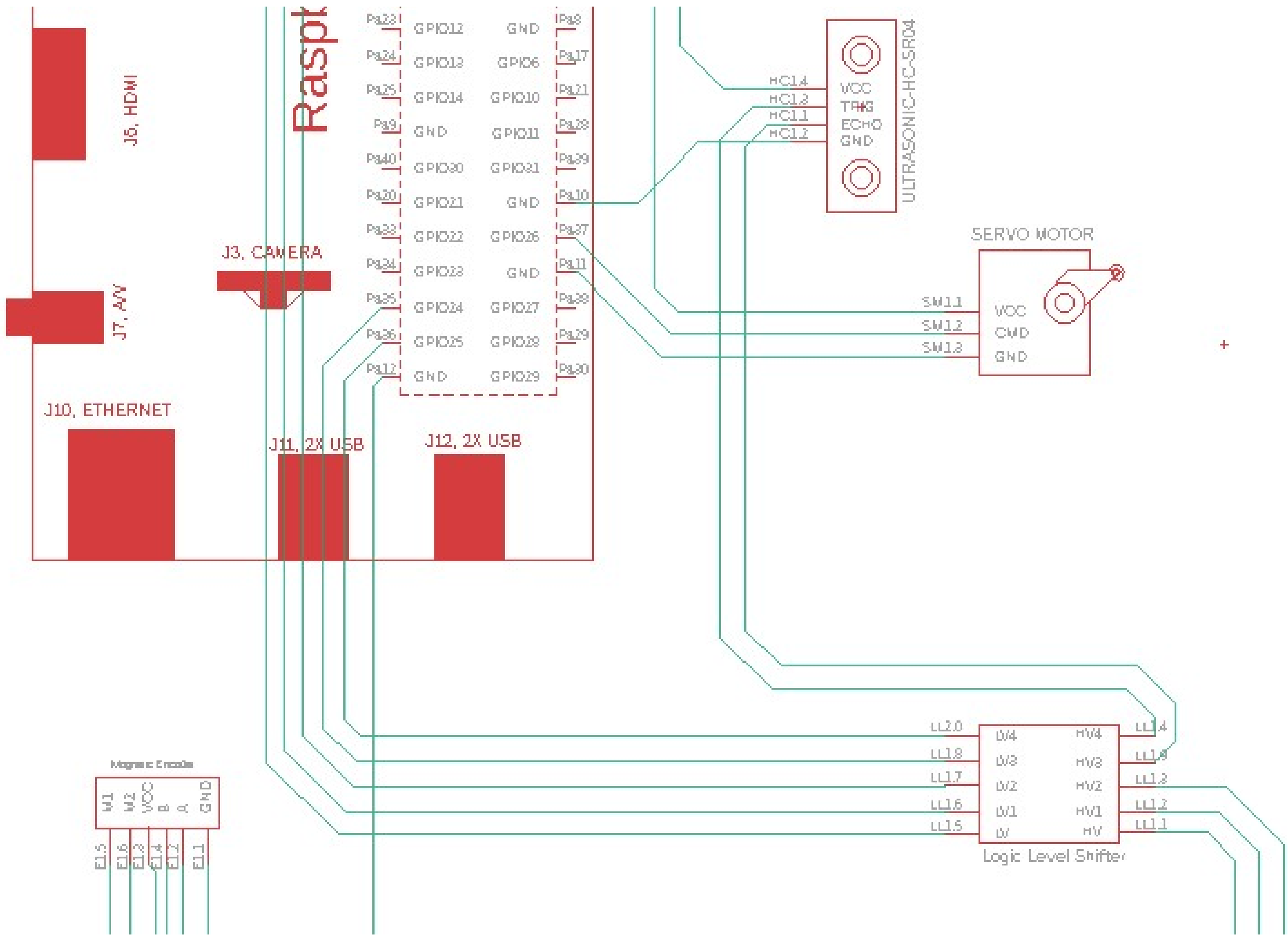


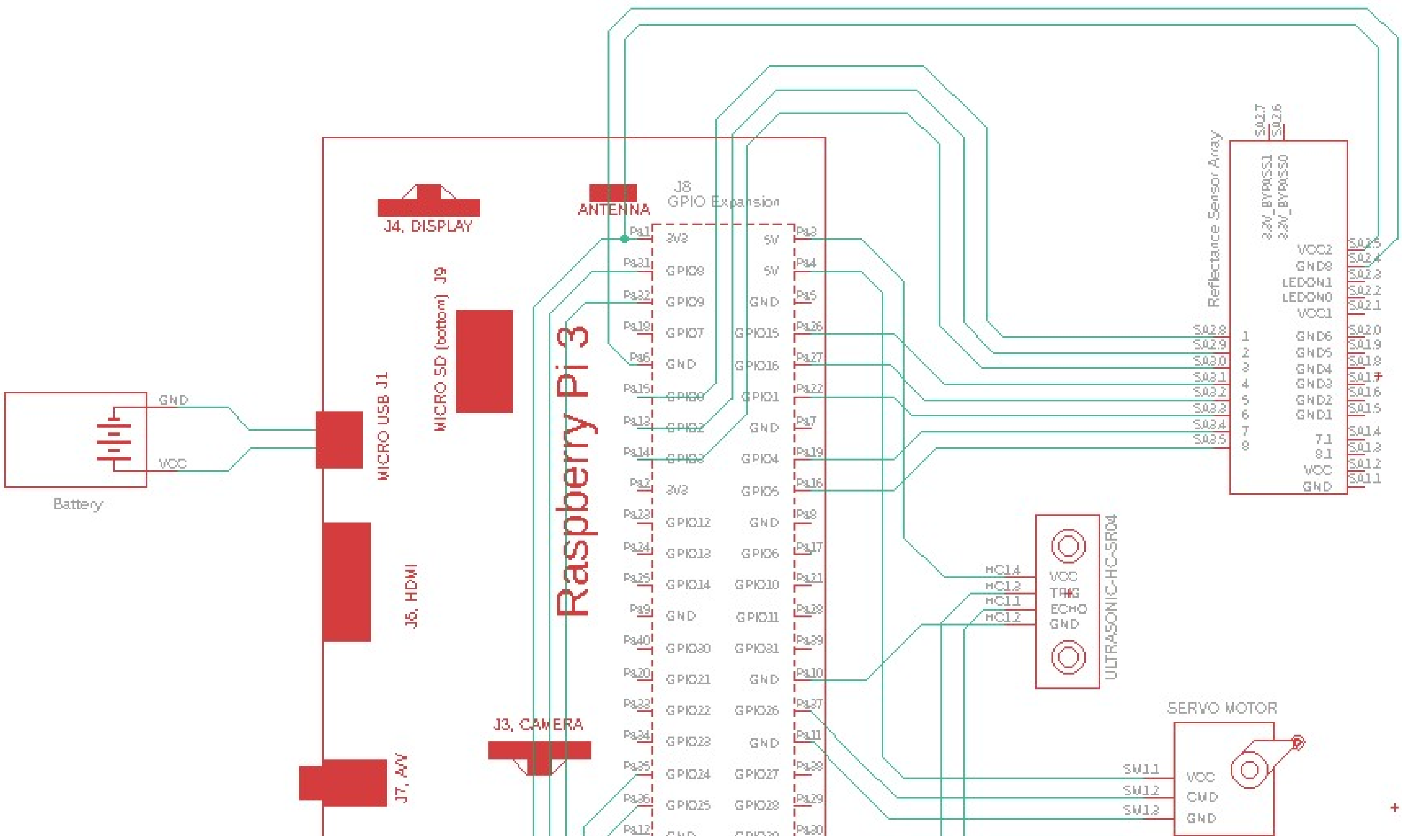
3/30/2019 8:22 PM f=0.75 C:\Users\Daniel\Documents\EAGLE\projects\astech_proj.sch (Sheet: 1/1)



LV4
LV3
LV2
LV1
LV
HV4
HV3
HV2
HV1
HV
Logic Level Shifter
SLS Battery
VCC
GND
ARDUINO
UNO
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
GND
AREF
SDA
SCL
A5
A4
A3
A2
A1
A0
VIN
GND1
GND2
5V
3V
/RESET
IOREF
RESET
USB
VCC
OUT
EN
GND
Analog Distance Sensor


Magnetic Encoder
W1 W2 VCC B A GND
E1.5 E1.6 E1.3 E1.4 E1.2 E1.1
Logic Level Shifter
LV4 LV3 LV2 LV1 LV
HV4 HV3 HV2 HV1 HV
LL2.0 LL1.8 LL1.7 LL1.6 LL1.5
LL1.4 LL1.9 LL1.3 LL1.2 LL1.1
SLS Battery
VCC
GND
M
GND
VCC
DCM1.2
DCM1.1
DC MOTOR
DC Motor Driver
MD1.8 MD1.6 MD1.7 MD2.0 MD1.5
PW W1 W2 VIN GND2
PWM DIR /FLT /EN GND1
MD1.9 MD1.11 MD1.3 MD1.2 MD1.4
ARDUINO
UNO
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 GND AREF SDA SCL
A5 A4 A3 A2 A1 A0 VIN GND1 GND2 5V 3V /RESET IOREF
RESET
SDA
SCL
PWR
USB
Analog Distance Sensor
AS1.1 AS1.2 AS1.3 AS1.4
VCC OUT EN GND
Digital Distance Sensor
OUT VIN GND
DS1.1 DS1.2 DS1.3


J5, HDMI
J7, A/V
J3, CAMERA
J10, ETHERNET
J11, 2X USB
J12, 2X USB
GPIO12
GPIO13
GPIO14
GND
GPIO30
GPIO21
GPIO22
GPIO23
GPIO24
GPIO25
GPIO6
GPIO10
GPIO11
GPIO31
GPIO26
GPIO27
GPIO28
GPIO29
ULTRASONIC-HC-SR04
VCC
TRIG
ECHO
GND
HC1.4
HC1.3
HC1.1
HC1.2
SERVO MOTOR
SM1.1
SM1.2
SM1.3
VCC
CMD
GND
Logic Level Shifter
LV4
LV3
LV2
LV1
LV
HV4
HV3
HV2
HV1
HV
LL2.0
LL1.9
LL1.7
LL1.6
LL1.5
LL1.4
LL1.3
LL1.2
LL1.1
M1
M2
VCC
B
A
GND
E1.5
E1.6
E1.3
E1.4
E1.2
E1.1



Battery
GND
VCC
MICRO USB J1
J4, DISPLAY
ANTENNA
J8
GPIO Expansion
MICRO SD (bottom) J9
Raspberry Pi 3
J5, HDMI
J7, A/V
J3, CAMERA
Reflectance Sensor Array
ULTRASONIC-HC-SR04
VCC
TRIG
ECHO
GND
SERVO MOTOR
VCC
CMD
GND

Block Functionality

The speed control block includes an Arduino UNO, a DC Motor Driver Carrier, an Encoder and a DC Motor. Arduino UNO is the brain of this block, through which it processes signals from the Encoder that monitor engine rotation. Arduino will modify the PWM signal transmitted to DC Motor Driver Carrier to increase or decrease engine speed.

There are 4 cases where this block is used:

- when avoiding an obstacle at 80 - 120 cm;

- if it is desired to stop in front of an obstacle whose appearance is spontaneous, located between 20-80 cm;

- if it is desired to track a target vehicle at a distance of 5-10 cm;

- where tracking is desired.

The obstacle / distance detection block implies the detection of an obstacle or a target vehicle and the calculation of the distance between the car and the vehicle. The ultrasonic sensor is used exclusively to detect the type of obstacle (which should be bypassed / in front of which it must stop). It emits a signal that returns to encounter an obstacle. Raspberry Pi calculates the distance to that object using the time the signal was emitted by the ultrasonic sensor and the time it was received. Distance measurement is done by the analog sensor with a maximum radius of 150 cm. It is powered by a 3.3V voltage supplied by Arduino. Arduino will calculate based on the signal provided by the sensor distance to obstacle.

Stop obstacle block is based on the fact that the obstacle detection / distance detection block detected an object located in a range of 20-80 cm, which means that the car must stop in front of that obstacle progressively decreasing the speed up to a limit distance 15 cm from the obstacle, where the travel speed will be 0 m / s. This task will be performed by the speed control unit ordered by Arduino

Avoidance obstacle block is also used by the obstacle detection / distance detection block to detect the obstacle type and the distance to it. When it reaches a distance of 50 cm from the obstacle, Arduino will command the speed control block for a progressive reduction of speed until the obstacle is overcome. When the car is 30 cm from the obstacle, Arduino will pass the obstacle bypass to Raspberry Pi. The actual detour is performed by the servomotor, ordered in turn by Raspberry Pi. At the start of the bypass procedure, the next line block will no longer command the servo until the line detection sensor comes out of a 5 second timeout period and will begin to feel again the presence of the line, the command being played on the following line.

The following line block has as its principal element a line array sensor. If he does not feel the presence of the line he will pass this to Raspberry Pi, who in turn through the level shift logic will pass to the speed control block to reduce the speed to a certain limit, and if the line presence is felt, the speed of travel increase until it reaches an average travel speed. Using the 8 integrated line sensors on the array sensor, Raspberry Pi will command the servo motor to maintain the line's direction.

The following target block. At the first occurrence of an obstacle / car, the ultrasonic sensor will transmit a signal through which Raspberry Pi will calculate the distance to that object / car and forward it to Arduino. Based on this distance, Arduino will calculate the time until it reaches about 30 cm before the obstacle / car. If, after the estimated time, the sensor transmits a greater distance from the object / machine, it results that the obstacle is actually a car to be tracked. In this case, the speed control block will increase the speed until it reaches a distance of 5-10 cm from the target car, and then maintain that distance.

TEST SPECIFICATION

Test case objectives

1. Line following:

- The vehicle must follow a black line on a white ground.

- The vehicle must continue to follow the line after the gap.

2. Obstacle avoidance:

- The vehicle must not collide with obstacles.

- The vehicle must break to standstill if the distance is between 200 mm and 800 mm.

- The vehicle must steer to avoid the obstacle if the distance is between 800 mm and 1200 mm.

3. Distance control:

- The vehicle must follow another vehicle - called target vehicle - without collision.

Test strategy

Unit test for each hardware component for to check if it works in optimal parameters and to detect any bugs.

DC Motor Test Cases:

- Powering the motor at 5V and analyzing engine behavior.

- Change the polarity of the power supply to check the reverse mode.

Encoder Test Cases:

-By feeding the encoder at 5V, we test if the output signals A and B give 5V too.

-Verify rotation by analyzing A and B phase.

- We test whether when the encoder is powered and the motor rotates clockwise, the output signal A is the first one that is powered at 5V, and when the engine is running in reverse mode, the output signal B is powered first.

- We test if the encoder reads the corresponding input signals by comparing them with the outputs of the devices with which it is connected

DC Motor Driver:

-Testing that when M1 is powered up, the motor works in the sensitivity of the clock, and when M2 is powered, the engine runs in reverse mode.

-We test the input signals by comparing them with the outputs from the Arduino board.

-We test if the M1 and M2 outputs give 5V when powered.

-We tested if the PWM input signal changes, the engine speed also changes.

Digital/Analog Distance Sensor:

- Testing if the sensor works correctly.

-We tested whether when it was recently started it works properly or needs to be calibrated and if so, how long the calibration is needed.

- We test the reading limits if the distance is less than it's minimum and when it is greater than the maximum range.

-We test how the sensor works for different colors. If the colors are darker, how well they appreciate the distance from when the colors of the obstacles are lighter.

-We test how quickly the sensor responds if the obstacle is at a close distance, but when the distance is greater.

- We test whether the input and output signals match the expected ones

Ultrasonic sensor:

- Testing if the sensor works correctly.

- We test the minimum and maximum limit of the sensor.

- We test whether the VCC and TRIG input signals are reading 5V and the output signal ECHO give 5V.

- We test whether interference may arise in increasing the frequency of ultrasound emission or noise.

-We test the viewing angle according to the distance.

Reflectance Array Sensor:

- Testing if the sensor works correctly.
- Testing if calibration is required.
- We test the optimal distance between the sensor and the line.
- We test the optimal number of sensors to perform the task.
- Testing whether the outputs at the line have the same voltage as the applied voltage.
- We test the smooth operation of the sensor under low light conditions.

Servomotor:

- We test if it responds properly depending on the signal applied to the input.
- We test the ratio of the rotation angle of the servomotor to the steering angle.

For Arduino and Raspberry Pi we test the outputs.