```
int AENA = 6;
int Apin1 = 4;
int Apin2 = 5;
int BENA = 3;
int Bpin1 = 2;
int Bpin2 = 7;
long time = 0;
const int trigPinLeft = 43;    // Define the trigger pin for the
ultrasonic sensor
const int echoPinLeft = 42;
const int trigPinCenter = 41;
const int echoPinCenter = 40;
const int trigPinRight = 39;
const int echoPinRight = 38;
const long interval = 200;  // Interval in milliseconds for distance
measurement and printing
bool left = false;
bool right = false;
bool center = false;
String numberint; //defined integer to store number after conversion
using namespace std;///////
#include <stdlib.h>
#include <ctype.h>
#include <RH_ASK.h>
#ifdef RH_HAVE_HARDWARE_SPI
#include <SPI.h> // Not actually used but needed to compile
#endif
//RH_ASK driver(2000, 11, 4, 5 );
RH_ASK driver;


void setup() {
  Serial.begin(9600);   // Initialize serial communication at 9600 baud
rate
  if (!driver.init())
         Serial.println("init failed");
  pinMode(AENA, OUTPUT);
  pinMode(Apin1, OUTPUT);
  pinMode(Apin2, OUTPUT);
  pinMode(BENA, OUTPUT);
```

```
  pinMode(Bpin1, OUTPUT);
  pinMode(Bpin2, OUTPUT);
  pinMode(trigPinLeft, OUTPUT);   // Set trigPin as an output
  pinMode(echoPinLeft, INPUT);    // Set echoPin as an input
  pinMode(trigPinCenter, OUTPUT);   // Set trigPin as an output
  pinMode(echoPinCenter, INPUT);    // Set echoPin as an input
  pinMode(trigPinRight, OUTPUT);   // Set trigPin as an output
  pinMode(echoPinRight, INPUT);    // Set echoPin as an input
}
void loop() {
  uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];
  uint8_t buflen = sizeof(buf);
  String rcv;
  String rcv2;
  String number;
  int numberint2 = 0;
  //recieve message from transmitter
  if (driver.recv(buf, &buflen)){
    for (int i = 0; i < buflen; i++) {
      rcv= rcv + (char)buf[i];//converts hexadecimal to alpha characters
      }
  }
  if (rcv == "square"){
    square();
    }
  if (rcv == "circle"){
    digitalWrite(Apin1, HIGH);
    digitalWrite(Apin2, LOW);
    digitalWrite(Bpin1, HIGH);
    digitalWrite(Bpin2, LOW);
    analogWrite(AENA, 248);
    analogWrite(BENA, 255);
    }
  if (rcv == "turn"){
    turnAround();
    }
  if (rcv == "distance"){
    time = 0;
    digitalWrite(Apin2, LOW);
    digitalWrite(Apin1, LOW);
```

```
    buflen = sizeof(buf); // Reset the buffer length for the second
reception
    rcv2 = ""; // Initialize rcv2
    while (!driver.recv(buf, &buflen)) {} // Wait until a new transmission
is received
    for (int i = 0; i < buflen; i++) {
      rcv2 += (char)buf[i]; // Append characters to rcv2
      }
    numberint = " ";
    for (int i = 0; i < buflen; i++) {
      if (isdigit(rcv2[i])) {
        numberint = numberint + rcv2[i];
        }
      }
      if (numberint.length() > 0) {
        numberint2 = numberint.toInt();
        time = map(numberint2, 15, 30, 10083, 20166);
        distanceFunc();
        }
  }
  if (rcv == "stop") {
    stop();
    }

  if (rcv == "2") {
    Serial.println("sjsjjs");
  while (!Serial.available()) {
  leftSensor();
  delay(100);
  centerSensor();
  delay(100);
  rightSensor();
  delay(100);

  if (left || center || right ) {
    stop();
    delay(1000);
    if (left) {
      turnRight();
      leftSensor();
```

```
      delay(100);
      centerSensor();
      delay(100);
      rightSensor();
      delay(100);
    }
    if (center) {
      turnAround();
      leftSensor();
      delay(100);
      centerSensor();
      delay(100);
      rightSensor();
      delay(100);
    }
    if (right){
      turnLeft();
      leftSensor();
      delay(100);
      centerSensor();
      delay(100);
      rightSensor();
      delay(100);
    }
  }
  else {
    drive();
  }
  }
  }
}
void leftSensor() {
    digitalWrite(trigPinLeft, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPinLeft, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPinLeft, LOW);

    // Measure the duration of the pulse received
    long duration = pulseIn(echoPinLeft, HIGH, 20000);
```

```
    float distance = duration * 0.0343 / 2;

    if (distance > 1 && distance < 40){
        left = true; // Use assignment operator to update the variable
        // Print the distance measured in centimeters
        Serial.print("Left Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
    } else {
        left = false; // Use assignment operator to update the variable
    }
}

void centerSensor() {
    digitalWrite(trigPinCenter, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPinCenter, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPinCenter, LOW);

    // Measure the duration of the pulse received
    long duration = pulseIn(echoPinCenter, HIGH, 20000);
    float distance = duration * 0.0343 / 2;

    if (distance > 1 && distance < 40){
        center = true; // Use assignment operator to update the variable
        // Print the distance measured in centimeters
        Serial.print("Center Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
    } else {
        center = false; // Use assignment operator to update the variable
    }
}

void rightSensor() {
    digitalWrite(trigPinRight, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPinRight, HIGH);
    delayMicroseconds(10);
```

```
    digitalWrite(trigPinRight, LOW);

    // Measure the duration of the pulse received
    long duration = pulseIn(echoPinRight, HIGH, 20000);
    float distance = duration * 0.0343 / 2;

    if (distance > 1 && distance < 40){
        right = true; // Use assignment operator to update the variable
        // Print the distance measured in centimeters
        Serial.print("Right Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
    } else {
        right = false; // Use assignment operator to update the variable
    }
}
void drive() {////////////////Power both motors to drive
  digitalWrite(Apin1, HIGH);
  digitalWrite(Apin2, LOW);
  digitalWrite(Bpin1, HIGH);
  digitalWrite(Bpin2, LOW);
  analogWrite(AENA, 250);
  analogWrite(BENA, 255);
  }

void stop() {////////////////Turn off Both motors
  digitalWrite(Apin2, LOW);
  digitalWrite(Apin1, LOW);
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, LOW);
  }

void turnLeft() {////////////////Power Motor A forward and
  digitalWrite(Apin2, LOW);//B reversed to turn 90 degrees
  digitalWrite(Apin1, HIGH);//counter clockwise
  digitalWrite(Bpin2, HIGH);
  digitalWrite(Bpin1, LOW);
  analogWrite(AENA, 255);
  analogWrite(BENA, 255);
  delay(250);
```

```
  stop();
  }

void turnRight() {////////////////Power Motor A forward and
  digitalWrite(Apin2, HIGH);//B reversed to turn 90 degrees
  digitalWrite(Apin1, LOW);//counter clockwise
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, HIGH);
  analogWrite(AENA, 255);
  analogWrite(BENA, 255);
  delay(250);
  stop();
  }
void turnAround() {////////////////Power Motor A forward and
  digitalWrite(Apin2, LOW);//B reversed to turn 90 degrees
  digitalWrite(Apin1, HIGH);//counter clockwise
  digitalWrite(Bpin2, HIGH);
  digitalWrite(Bpin1, LOW);
  analogWrite(AENA, 255);
  analogWrite(BENA, 255);
  delay(500);
  stop();
  }

void square() {//////////////Drive HERBERT in a square
  drive();
  turnRight();
  drive();
  turnRight();
  drive();
  turnRight();
  drive();
  finished();
  stop();
  }
void distanceFunc() {////////Power both motors
  Serial.println(time);
  digitalWrite(Apin2, LOW);//for time required to meet
  digitalWrite(Apin1, HIGH);///distance requirement
  digitalWrite(Bpin2, LOW);
```

```
  digitalWrite(Bpin1, HIGH);
  analogWrite(AENA, 255);
  analogWrite(BENA, 255);
  delay(2000);
  Serial.println("time: ");
  Serial.println(time);
  time = 0;
  stop();
  }
void finished(){
  digitalWrite(Apin2, HIGH);
  digitalWrite(Apin1, LOW);
  analogWrite(AENA, 255);
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, HIGH);
  analogWrite(BENA, 255);
  delay(5000);
  }
```