#include <Wire.h>

#include <Keypad.h>

#include <NewPing.h>

/\*Hero Jr Control - Arduino Mega

Commands so far:

key() - read a button push from the keypad

ping() - read distance from ping sensor

led(#, state) - turn on led # 0 to 7 state is on or off

forward(distance, speed) - move forward so far at a speed

backward(distance, speed) - move backward so far at a speed

left(degree) - turn to the left

right(degree) - turn to the right

light() - read the light sensor

volt() - check the voltage levels for the drive battery

test() - don't know yet

jrinit() - don't know yet

\*\*\* Voltage Sensor Information

This seems to work 3.3v hooked to output side

and "S" hooked to Analog input 1

It qives whole numbers as output

Tested on the UNO, and not sure why it needs "Wire.h"

\*/

int lightPin = 0; //analog pin0 for photoresistor

#define TRIGGER\_PIN 13 // Arduino pin tied to trigger pin on the ultrasonic sensor.

#define ECHO\_PIN 12 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define MAX\_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm. About 78 Inches

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE); // NewPing setup of pins and maximum distance.

int val11;

int val2;

int led0 = 22;

int led1 = 24;

int led2 = 26;

int led3 = 28;

int led4 = 30;

int led5 = 32;

int led6 = 34;

int led7 = 36;

int state0 = 1;

int state1 = 1;

int state2 = 1;

int state3 = 1;

int state4 = 1;

int state5 = 1;

int state6 = 1;

int state7 = 1;

const byte ROWS = 4; // Four rows

const byte COLS = 4; // four columns

// Define the Keymap

char keys[ROWS][COLS] = {

{'C','D','E','F'},

{'8','9','A','B'},

{'4','5','6','7'},

{'0','1','2','3'}

};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte rowPins[ROWS] = { 4, 5, 6, 7 };

// Connect keypad COL0, COL1 and COL2 to these Arduino pins.

byte colPins[COLS] = { 8, 9, 10, 11 };

// Create the Keypad

Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

int number;

int state;

String command;

int distance;

int speed;

int degree;

int fl;

void setup()

{

Serial.begin(9600);

pinMode(led0, OUTPUT);

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

pinMode(led5, OUTPUT);

pinMode(led6, OUTPUT);

pinMode(led7, OUTPUT);

digitalWrite(led0, 1);

digitalWrite(led1, 1);

digitalWrite(led2, 1);

digitalWrite(led3, 1);

digitalWrite(led4, 1);

digitalWrite(led5, 1);

digitalWrite(led6, 1);

digitalWrite(led7, 1);

}

void loop()

{

fl = 0;

if (Serial.available() >0)

{

command = Serial.readStringUntil(';');

if (command == "led") {

number = Serial.parseInt();

led (number);

fl = 1;

}

if (command == "key") { key(); fl = 1; }

if (command == "ping") { ping(); fl = 1; }

if (command == "forward") { distance = Serial.parseInt();

speed = Serial.parseInt();

forward(distance, speed);

fl = 1;

}

if (command == "backward") { distance = Serial.parseInt();

speed = Serial.parseInt();

backward(distance, speed);

fl = 1;

}

if (command == "left") { degree = Serial.parseInt();

left(degree);

fl = 1;

}

if (command == "right") { degree = Serial.parseInt();

right(degree);

fl = 1;

}

if (command == "light") { light(); fl = 1;}

if (command == "volt") { volt(); fl = 1;}

if (command == "test") { test(); fl = 1;}

if (command == "jrinit") { jrinit(); fl = 1;}

if (fl == '0') {

Serial.println("ERR");

fl = 0;

}

}

}

void key() {

delay(100);

char key = kpd.getKey();

Serial.println(key);

}

void ping() {

unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).

Serial.println(uS / US\_ROUNDTRIP\_IN); // Convert ping time to distance in cm and print result (0 = outside set distance range)

}

void led(int number1) {

switch (number1) {

case 0:

state0 = !state0;

digitalWrite(led0, state0);

break;

case 1:

state1 = !state1;

digitalWrite(led1, state1);

break;

case 2:

state2 = !state2;

digitalWrite(led2, state2);

break;

case 3:

state3 = !state3;

digitalWrite(led3, state3);

break;

case 4:

state4 = !state4;

digitalWrite(led4, state4);

break;

case 5:

state5 = !state5;

digitalWrite(led5, state5);

break;

case 6:

state6 = !state6;

digitalWrite(led6, state6);

break;

case 7:

state7 = !state7;

digitalWrite(led7, state7);

break;

}

}

void forward(int distance, int speed) {

}

void backward(int distance, int speed) {

}

void left(int degree) {

}

void right(int degree) {

}

void light() {

int photor = analogRead(lightPin);

Serial.println(photor);

delay(50);

}

void volt() {

float temp;

val11=analogRead(1);

temp=val11/4;

val11=(int)temp;

val2=((val11)/10);

Serial.println(val2);

}

void test() {

}

void jrinit() {

}