Ardruino Mega - PCA9685 PWM Servo Driver Example

https://forum.dronebotworkshop.com/arduino/programming-the-arduino-mega-with-pca9685/

一張含有 文字, 圖表, 電子產品, 電子工程 的圖片

自動產生的描述

/\*

Ardruino Mega - PCA9685 PWM Servo Driver Example

\*/

// Include Wire Library for I2C Communications

#include <Wire.h>

// Include Adafruit PWM Library

#include <Adafruit\_PWMServoDriver.h>

#define MIN\_PULSE\_WIDTH 650

#define MAX\_PULSE\_WIDTH 2350

#define FREQUENCY 50

Adafruit\_PWMServoDriver pwm = Adafruit\_PWMServoDriver();

// Define Potentiometer Inputs

int controlA = A0;

int controlB = A1;

int controlC = A2;

int controlD = A3;

int controlE = A4;

int controlF = A5;

int controlG = A6;

int controlH = A7;

int controlI = A8;

int controlJ = A9;

int controlK = A10;

int controlL = A11;

int controlM = A12;

int controlN = A13;

int controlO = A14;

int controlP = A15;

// Define Motor Outputs on PCA9685 board

int motorA = 0;

int motorB = 1;

int motorC = 2;

int motorD = 3;

int motorE = 4;

int motorF = 5;

int motorG = 6;

int motorH = 7;

int motorI = 8;

int motorJ = 9;

int motorK = 10;

int motorL = 11;

int motorM = 12;

int motorN = 13;

int motorO = 14;

int motorP = 15;

void setup()

{

pwm.begin();

pwm.setPWMFreq(FREQUENCY);

}

void moveMotor(int controlIn, int motorOut)

{

int pulse\_wide, pulse\_width, potVal;

// Read values from potentiometer

potVal = analogRead(controlIn);

// Convert to pulse width

pulse\_wide = map(potVal, 0, 1023, MIN\_PULSE\_WIDTH, MAX\_PULSE\_WIDTH);

pulse\_width = int(float(pulse\_wide) / 1000000 \* FREQUENCY \* 4096);

//Control Motor

pwm.setPWM(motorOut, 0, pulse\_width);

}

void loop() {

//Control Motor A

moveMotor(controlA, motorA);

//Control Motor B

moveMotor(controlB, motorB);

//Control Motor C

moveMotor(controlC, motorC);

//Control Motor D

moveMotor(controlD, motorD);

//Control Motor E

moveMotor(controlE, motorE);

//Control Motor F

moveMotor(controlF, motorF);

//Control Motor G

moveMotor(controlG, motorG);

//Control Motor H

moveMotor(controlH, motorH);

//Control Motor I

moveMotor(controlI, motorI);

//Control Motor J

moveMotor(controlJ, motorJ);

//Control Motor K

moveMotor(controlK, motorK);

//Control Motor L

moveMotor(controlL, motorL);

//Control Motor M

moveMotor(controlM, motorM);

//Control Motor N

moveMotor(controlN, motorN);

//Control Motor O

moveMotor(controlO, motorO);

//Control Motor P

moveMotor(controlP, motorP);

}