Robotics & Automation

DIGITALWEIGHT KIT

USER MANUAL

www.svrinfotech.net

Table of Contents:

1. Introduction
2. General Precautions
3. Components
    a. Mechanical Components
    b. Electrical Components
    c. Prerequisites
4. Connections
5. Circuit Diagram
6. Code Explanation
7. Warranty Terms & Conditions

Introduction:

In Education Module, SVR InfoTech offers DIGITAL WEIGHT Kit. This application is simple just by using lode cell with an Arduino kit to display with lcd 16,2 respect to the weight read by the microcontroller which in this case is the Arduino . The lode cell is connected with the Arduino with an digital input pin D4,D5.it use a function map in the Arduino IDE.

General Precautions:

Caution: To avoid injury, damage to the robot or equipment, please follow the provided guidelines.

1. Keep away from pets and animals of any kind, animals may behave erratically in the presence of the robot.
2. If the robot is operating abnormally, there is an unusual sound, smell or smoke is detected:
    a. Turn off the robot immediately
3. Always follow the installation and service instructions closely. Keep manuals for future reference.
4. This guide does not cover all possible safety issues or conditions. Always use common sense and good judgment.
5. Please take care of this unit and its accessories, keep them clean. Please do not let this unit or accessories exposed to fire/burning cigarettes, etc... Try to keep the robot and its accessories dry; please do not let this unit exposed to water or moisture.
6. Please do not break, throw or trample the robot.
7. Avoid installation in extremely hot, rainy or water splashing, or being placed in high temperature or moist environment.
8. Please use the accessories provided with this robot.

Components:

ELECTRICAL COMPONENTS		
SR. NO.	PART NAME	QUANTITY
1	Arduino Uno	1
2	10kg load cell	1
3	16x2 LCD display	1
4	hx711 weight sensor	1

PREREQUISITES		
SR. NO.	PART NAME	QUANTITY
1	Personal Computer with Arduino IDE	1
2	USB B to USB A cable	1

Connections:

For load cell Sensor and **Arduino**		
SR. NO.	Sensor pin	Arduino Pins
1	HX711 dout pin	D4
2	HX711 SCK pin	D5

Code Explanation:

Arduino PIN declaration.

```
//pins:
const int HX711_dout = 4; //mcu > HX711 dout pin
const int HX711_sck = 5; //mcu > HX711 sck pin

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_eepromAdress = -13;
unsigned long t = 0;
```

Void setup for digital weight kit

```
  Serial.begin(57600); delay(10);
  Serial.println();
  Serial.println("Starting...");
  LoadCell.begin();
  //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to
positive
  float calibrationValue; // calibration value (see example file
"Calibration.ino")
  calibrationValue = 250.0; // uncomment this if you want to set the
calibration value in the sketch
#if defined(ESP8266)|| defined(ESP32)
  //EEPROM.begin(512); // uncomment this if you use ESP8266/ESP32 and want to
fetch the calibration value from eeprom
#endif
  //EEPROM.get(calVal_eepromAdress, calibrationValue); // uncomment this if you
want to fetch the calibration value from eeprom
 lcd.backlight();
  unsigned long stabilizingtime = 2000; // preciscion right after power-up can
be improved by adding a few seconds of stabilizing time
  boolean _tare = true; //set this to false if you don't want tare to be
performed in the next step
  LoadCell.start(stabilizingtime, _tare);
  lcd.init();

  if (LoadCell.getTareTimeoutFlag()) {
    Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
```

```
    while (1);
  }
  else {
    LoadCell.setCalFactor(calibrationValue); // set calibration value (float)
    Serial.println("Startup is complete");
  }
}
```

Check the weight , and display on the LCD.

```
 if (newDataReady) {
  if (millis() > t + serialPrintInterval)
  {
    float I = LoadCell.getData();
    Serial.print ("Load cell output Val: ");
    Serial.println(I);
     lcd.setCursor(1, 1); // set cursor to secon row
     lcd.print("Weight= ");
     lcd.print(I, 1);
     lcd.print(" g ");
    newDataReady = 0;
    t = millis();
  }
```

Code:

```
/*

   ---------------------------------------------------------------------------
-------

   HX711_ADC

   Arduino library for HX711 24-Bit Analog-to-Digital Converter for Weight Scales

   Olav Kallhovd sept2017

   ---------------------------------------------------------------------------
-------

*/


/*
```

```
   Settling time (number of samples) and data filtering can be adjusted in the
config.h file

   For calibration and storing the calibration value in eeprom, see example file
"Calibration.ino"



   The update() function checks for new data and starts the next conversion. In
order to acheive maximum effective

   sample rate, update() should be called at least as often as the HX711 sample
rate; >10Hz@10SPS, >80Hz@80SPS.

   If you have other time consuming code running (i.e. a graphical LCD), consider
calling update() from an interrupt routine,

   see example file "Read_1x_load_cell_interrupt_driven.ino".



   This is an example sketch on how to use this library

*/

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <HX711_ADC.h>

#if defined(ESP8266)|| defined(ESP32) || defined(AVR)

#include <EEPROM.h>

#endif


//pins:

const int HX711_dout = 4; //mcu > HX711 dout pin

const int HX711_sck = 5; //mcu > HX711 sck pin


//HX711 constructor:

HX711_ADC LoadCell(HX711_dout, HX711_sck);


const int calVal_eepromAdress = -13;

unsigned long t = 0;
```

```
void setup() {

  Serial.begin(57600); delay(10);

  Serial.println();

  Serial.println("Starting...");

  LoadCell.begin();

  //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to positive

  float calibrationValue; // calibration value (see example file "Calibration.ino")

  calibrationValue = 250.0; // uncomment this if you want to set the calibration value in the sketch

#if defined(ESP8266)|| defined(ESP32)

  //EEPROM.begin(512); // uncomment this if you use ESP8266/ESP32 and want to fetch the calibration value from eeprom

#endif

  //EEPROM.get(calVal_eepromAdress, calibrationValue); // uncomment this if you want to fetch the calibration value from eeprom

 lcd.backlight();

  unsigned long stabilizingtime = 2000; // preciscion right after power-up can be improved by adding a few seconds of stabilizing time

  boolean _tare = true; //set this to false if you don't want tare to be performed in the next step

  LoadCell.start(stabilizingtime, _tare);

  lcd.init();


  if (LoadCell.getTareTimeoutFlag()) {

    Serial.println("Timeout, check MCU>HX711 wiring and pin designations");

    while (1);

  }

  else {
```

```
    LoadCell.setCalFactor(calibrationValue); // set calibration value (float)
    Serial.println("Startup is complete");
  }
}

void loop() {
  lcd.clear();
   int m=Serial.available();
    while (Serial.available() > 0) {
      //lcd.clear();
      lcd.write(Serial.read());

  delay(1000);
}
  lcd.setCursor(2, 0);
   lcd.print("SVR INFOTECH");
  static boolean newDataReady = 0;
  const int serialPrintInterval = 0; //increase value to slow down serial print
activity

  // check for new data/start next conversion:
  if (LoadCell.update()) newDataReady = true;

  // get smoothed value from the dataset:
  if (newDataReady) {
   if (millis() > t + serialPrintInterval)
   {
      float I = LoadCell.getData();
      Serial.print ("Load cell output Val: ");
```

```
      Serial.println(I);
       lcd.setCursor(1, 1); // set cursor to secon row
       lcd.print("Weight= ");
       lcd.print(I, 1);
       lcd.print(" g ");
      newDataReady = 0;
      t = millis();
    }
  }

  // receive command from serial terminal, send 't' to initiate tare operation:
  if (Serial.available() > 0) {
    char inByte = Serial.read();
    if (inByte == 't') LoadCell.tareNoDelay();
  }

  // check if last tare operation is complete:
  if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare complete");

}
}
```

Warranty Terms and Conditions:

Warranty Period: - 90 Days from the date of delivery.

What is covered: - Any Technical defect, malfunctioning.

What is not covered: - Physical Damage, Water Damage, Wear & Tear.

What will we do: - Repair or Replacement whichever will be applicable.

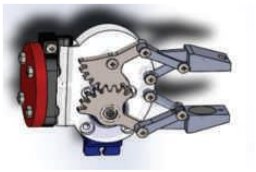
Other terms and conditions: -

- Warranty will be void if warranty seals are broken.
- Warranty will be void if the product is mishandled.
- Warranty will be void on installation to wrong voltage, overload and wrong application.
- Use other than in accordance to handling instructions.
- Warranty will not extend after replacement.

DIGITAL WEIGHT KIT

OUR OTHER PRODUCTS

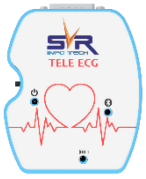
Hast Articulated Robot



Miniature Geared Gripper

Miniature Cam Guided Gripper

Miniature Curvilinear Gripper



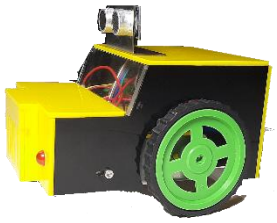
Tele ECG

SCARA Robot

Flexible Robotic System

Robotic leech

Self-Balancing Robot



Ultrasonic Scanner

Maze Robot

Digital Dice kit

Multi Gripper Robot

Conveyor Belt object counter

Line Follower Robot System

Parallel Manipulator Gripper stem

Amber Plaza, 3rd Floor, Near Bank of Maharashtra, Sinhgad College Rd, Ambegaon Budruk, Pune-411046.

Website**: www.svrinfotech.net** E-mail: **admin@svrinfotech.net / support@svrinfotech.net**

+91 9923444362, +91 9975507755 , 020 2984 0013