智能仓储系统的开发研究

先进计算与机器人研究所

2023 年 7 月 4 日

目录

1 第七章：部署

这一章主要介绍实际如何部署好一个货架。一个完整的货架包含：一个触摸屏，多个电子秤还有整个架子。软件上需要给所有电子秤烧录好代码，触摸屏的 UI 界面设计好；硬件上要连好 Arduino 和压力传感器，读卡器等元器件的连线，以及整个货架的 IIC 通信线路。上述过程分为四步来完成：

- 第一步：秤的组装；
- 第二步：上传程序；
- 第三步：货架布线；
- 第四步：触摸屏显示；

1.1 秤的组装

压力测量模块连接，详见第一章：

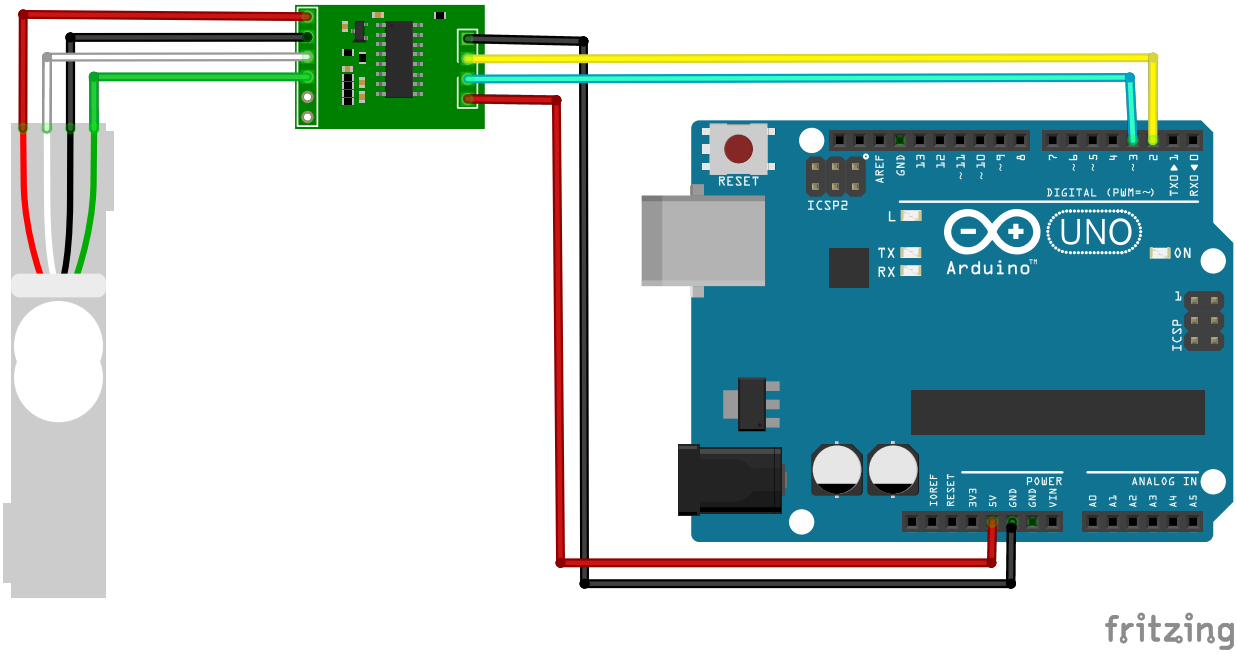

图 1：压力测量模块流程

两种读卡器连接，详见第五章第六章：

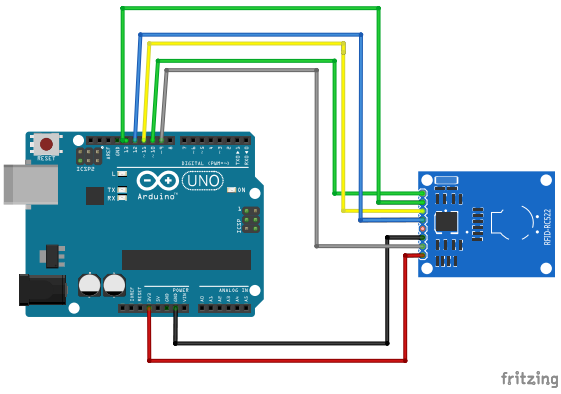

图 2：RC522 硬件连接

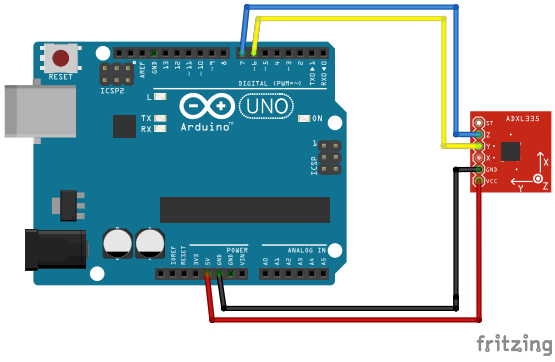

图 3：R505 与 arduino 的实物连接图

1.2 上传程序

这里我们选择的是 TCA9548A 的布线方式, arduino 分为多个主机一个从机,

主机: 如果是采取近的读卡器读取物品种类及单个物体质量信息, 就烧录下面的程序:

```
#include "master.h"
#include "transform.h"
#include "Surface.h"
#include "Calibrate.h"
#include "Oled.h"
#include "RC522.h"

master m1;
Surface YL_Surface;
Calibrate YL_Calibrated;
transform tf;
Oled oled;
RC522 rc522;

long numbefore=0, numnow=1;
char te[3];
unsigned long Sweight;
bool rflag=1; //1 ---> read; 0 ---> write

void setup() {
  Serial.begin(9600);
  m1.initialize(8); //8needschanged
  YL_Calibrated.setpin_SCKDT(4, 5);
  YL_Calibrated.set_range(20);
  YL_Calibrated.kb_Initialize();
  oled.initialize();
  rc522.initialize(9,10);
}

void loop() {
  switch(rflag){
    case 0:
      while(1) {
        bool state = rc522.write();
        if(state==1)
          break;
      }
      rflag=1;
      break;
    case 1:
      while(1) {
        bool state = rc522.read();
        if(state==1)
          break;
      }
      break;
  }
  for (int i=0; i<sizeof(rc522.Type_Name);i++) te[i]=rc522.Type_Name[i];
  Sweight=rc522.Single_Weight;

  numbefore = numnow;
  unsigned long CalibratedWeight = YL_Calibrated.Output_CalibratedWeight(YL_Surface.
      Get_Surface());
  numnow = ceil(CalibratedWeight/Sweight);

  bool flag= (numbefore==numnow?0:1);
```

```
    if(flag){
    tf.initialize(te, m1.address, numnow, CalibratedWeight);
    tf.pack();
    digitalWrite(3,HIGH);
    m1.send(9, tf.Transmission_Information);
    Serial.println(tf.Transmission_Information);
    oled.showIIC(te, numnow);
    digitalWrite(3,LOW);
    }

    delay(3000);
  }
```

如果是采取远的读卡器读取物品种类及单个物体质量信息, 就烧录下面的程序:

```
#include "master.h"
#include "transform.h"
#include "Surface.h"
#include "Calibrate.h"
#include "Oled.h"
#include "RC522.h"
#include "RC505.h"

master m1;
Surface YL_Surface;
Calibrate YL_Calibrated;
transform tf;
Oled oled;
RC522 rc522;
RC505 rc505;

long numbefore=0, numnow=1;
char te[3];
unsigned long Sweight;
char MessageNow[arrayMax];
bool flag=0; //1--->write; 0--->read

void setup() {
  //设置串口波特率38400
  Serial.begin(38400);
  m1.initialize(8); //8needschanged
  YL_Calibrated.setpin_SCKDT(4, 5);
  YL_Calibrated.set_range(20);
  YL_Calibrated.kb_Initialize();
  oled.initialize();
  rc522.initialize(9,10);
}


void loop() {
  switch(flag){
    case 0:
      // while(1) {
      //  bool state = rc505.write();
      //  if(state==1)
      //    break;
      // }
      rflag=1;
      // break;
    case 1:
      while(1) {
        bool state = rc505.read(MessageNow);
```

```
          if(state==1)
            break;
        }
        break;
    }
    for (int i=0; i<sizeof(rc505.Type_Name);i++) te[i]=rc505.Type_Name[i];
    Sweight=rc505.Single_Weight;

    numbefore = numnow;
    unsigned long CalibratedWeight = YL_Calibrated.Output_CalibratedWeight(YL_Surface.
        Get_Surface());
    numnow = ceil(CalibratedWeight/Sweight);

    bool flag= (numbefore==numnow?0:1);
    if(flag){
    tf.initialize(te, m1.address, numnow, CalibratedWeight);
    tf.pack();
    digitalWrite(3,HIGH);
    m1.send(9, tf.Transmission_Information);
    Serial.println(tf.Transmission_Information);
    oled.showIIC(te, numnow);
    digitalWrite(3,LOW);
    }

    delay(3000);
  }
```

从机 (也就是与 TCA9548A 相连的 arduino 板上的主程序如下):

```
#include "Oled.h"
#include "slave.h"
#include "transform.h"

String slave::comdata="";

Oled oled;
slave s;
transform tf;

void setup() {
  Serial.begin(9600);
  s.initialize(8);
  oled.initialize();
  pinMode(7,INPUT);
  digitalWrite(7, LOW);
}

void loop() {
  tf.unpack(s.comdata);
  oled.showIIC(tf.Type_Name, tf.Number, tf.DEVICE_ADDRESS);
  delay(1000);
}
```

1.3 货架布线

IIC 通信及 OLED 屏显示连线, 详见第三章第四章:

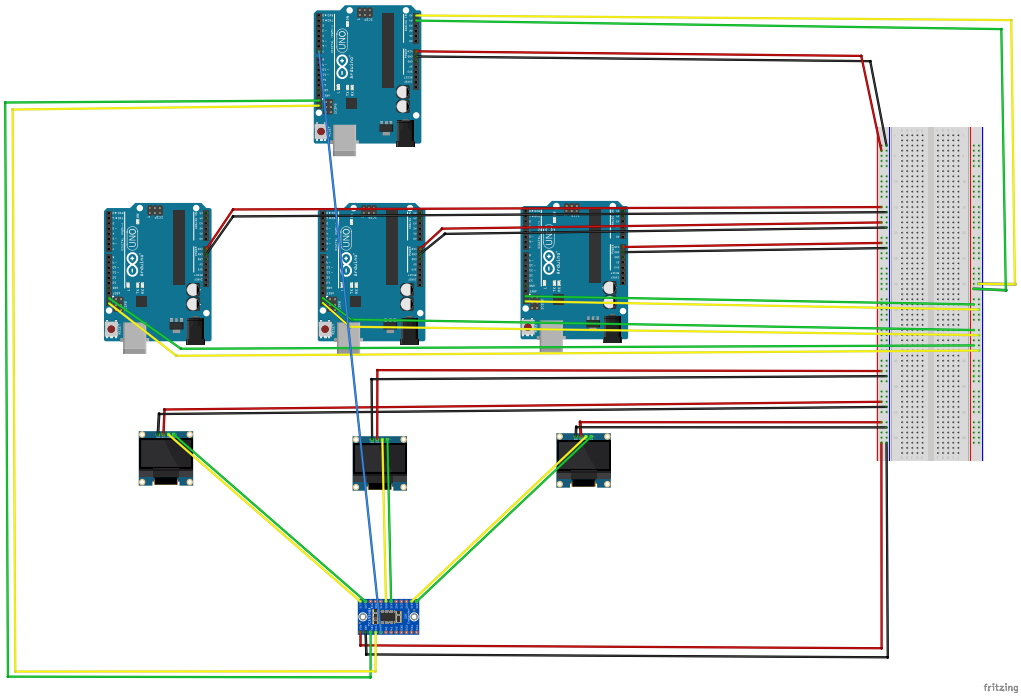

图 4: 一层货架示意图

对于第二层甚至更多层的话, 就要考虑总的电子秤数量是否超过 8, 超过了的话则需要更多从机和 TCA9548A 模块。

1.4 触摸屏显示

上传下面的 python 代码到树莓派上运行, 树莓派就可以接收到从机发来的信号, 从而在屏幕上显示对应电子秤的重量变化。

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout,
    QGridLayout, QLabel, QMainWindow
from PyQt5.QtGui import QPixmap
from PyQt5.QtCore import Qt, QTimer, QIODevice, QByteArray
from PyQt5.QtSerialPort import QSerialPort, QSerialPortInfo
from qt_material import apply_stylesheet
from PyQt5.uic import loadUi


class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        loadUi("main_window1.ui", self) # 加载UI文件
        # 设置窗口标题和大小
        self.setWindowTitle('货物管理')

        # 创建网格布局
        grid = QGridLayout(self)
        grid.setAlignment(Qt.AlignCenter)

        pixmap = QPixmap('jimu1.jpg')
        pixmap = pixmap.scaled(150, 130, Qt.KeepAspectRatio)
        self.label1.setPixmap(pixmap)
```

```
        self.label1.setFixedSize(pixmap.width(), pixmap.height())
        self.label1.setAlignment(Qt.AlignCenter)
        self.pushButton.clicked.connect(self.click_button)
        self.pushButton_2.clicked.connect(self.click_button2)

        # 应用Qt Material主题
        apply_stylesheet(self, theme='light_teal.xml')

        # 创建串口监视器
        self.serial = QSerialPort()
        self.serial.setPortName("COM3") # 设置端口号
        self.serial.setBaudRate(QSerialPort.Baud9600)
        self.serial.readyRead.connect(self.handle_serial_data)
        self.serial.open(QIODevice.ReadWrite)

    def click_button(self):
        self.serial.close()

    def click_button2(self):
        self.serial = QSerialPort()
        self.serial.setPortName("COM8") # 设置端口号
        self.serial.setBaudRate(QSerialPort.Baud9600)
        self.serial.readyRead.connect(self.handle_serial_data)
        self.serial.open(QIODevice.ReadWrite)

    def handle_serial_data(self):
        # 读取串口数据
        data = self.serial.readAll().data()
        data_str = str(data, encoding='utf-8')
        print(data)
        # 处理数据
        module_data = data_str.split(',')
        for i, module_str in enumerate(module_data):
            # 如果模块数据不为空，则更新数量和种类
            if module_str:
                try:
                    type_, num = module_str.split(':')
                    self.update_module_info(i, type_, num)
                except ValueError as e:
                    print(f"Error: {e}. Received data: {module_str}")

    def update_module_info(self, module_num, type_, num):
        # 获取模块信息字典
        module_info = {
            0: {'num': None, 'type': 'AA', 'image': 'jimu1.jpg'},
            1: {'num': None, 'type': 'B', 'image': 'picture2.jpg'},
            2: {'num': None, 'type': 'C', 'image': 'picture3.jpg'},
            3: {'num': None, 'type': 'D', 'image': 'picture4.jpg'},
            4: {'num': None, 'type': 'E', 'image': 'picture5.jpg'},
            5: {'num': None, 'type': 'F', 'image': 'picture6.jpg'},
            6: {'num': None, 'type': 'G', 'image': 'picture7.jpg'},
            7: {'num': None, 'type': 'H', 'image': 'picture8.jpg'},
            8: {'num': None, 'type': 'I', 'image': 'picture9.jpg'},
            9: {'num': None, 'type': 'J', 'image': 'picture10.jpg'},
            10: {'num': None, 'type': 'K', 'image': 'picture11.jpg'},
            11: {'num': None, 'type': 'L', 'image': 'picture12.jpg'}
        }

        # 更新模块信息字典中对应模块的数量和种类
        for i in range(12):
            if module_info[i]['type'] == type_:
```

```
                module_info[i]['num'] = num
                num_label = self.findChild(QLabel, f'label_num_{i}')
                type_label = self.findChild(QLabel, f'label_type_{i}')
                if num_label is None:
                    print(f"label_num_{module_num} not found!")
                    return

                num_label.setText(num)
                break


if __name__ == '__main__':
    app = QApplication(sys.argv)
    main_window = MainWindow()
    main_window.show()
    sys.exit(app.exec_())
```