**陀螺仪**

一、任务要求

（1）查阅JY62相关资料，了解工作参数；

（2）与上位机连接调试；

（3）设计程序，用单片机读取三轴信号并显示在LCD上；

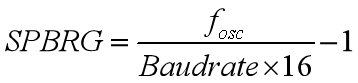
二、实验原理

（1）串口通信

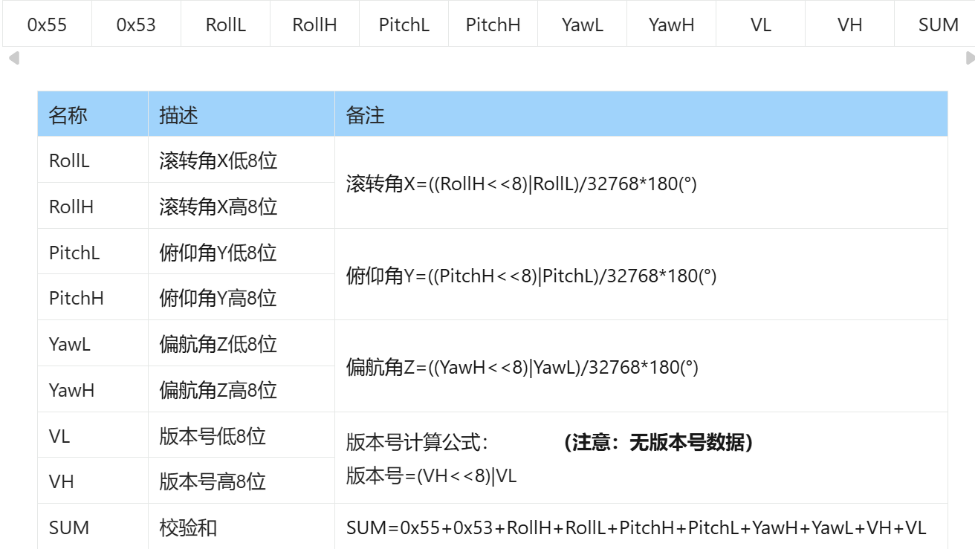


查阅JY62资料，可知串口通信使用的是UART，串口速率有两种可选。此处选用9600波特率，明显误差小得多，UART是异步通信协议，对波特率高度严格要求。实际通信中，如果波特率误差在3%以上，8位数据帧会偏移>1.5位，就有可能造成通信失败，在此9600是优选。

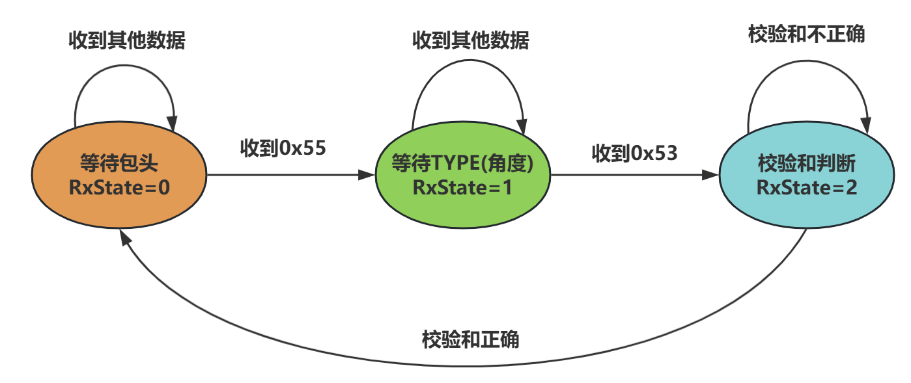
波特率计算公式如下：



（2）HEX数据包接收



根据发来的一帧数据，可对通过包头以及校验和判断发来的数据是否正确。基于此，使用状态机的方式来接收和判断数据，如图所示。



（3）原始数据处理







（4）180°到-180°突变处理

new\_yaw = 180 - (180 - yaw) % 360

1. 偏移处理：首先将角度偏移180度，这样原来的-180°到180°范围就变成了 0° 到 360°。
2. 模运算：使用模360运算将角度限制在 0°到 360°之间，处理了角度超出范围的情况。
3. 反向偏移：将结果再偏移回原来的范围，即从 0°到 360°变回-180°到180°。

（5）C语言数据类型



陀螺仪原始数据为十六进制并拆分为高低字节，即uint8\_t，需将其转换成int16\_t或uint16\_t才能合并，进行正常计算。

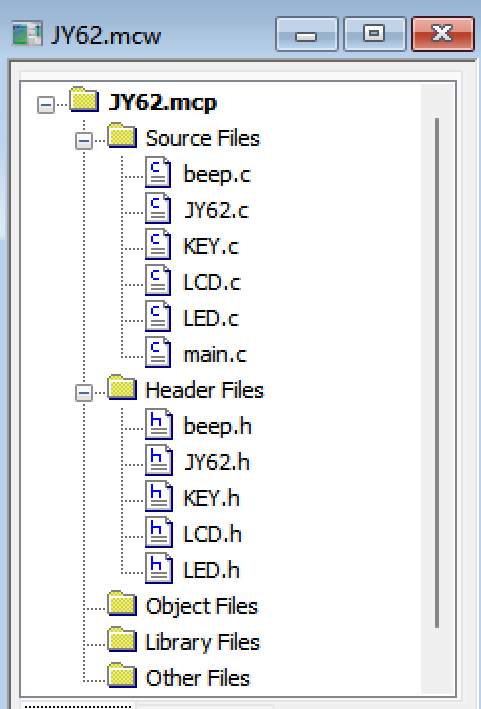
三、工程实现

工程代码已放到本人github账号并开源:

<https://github.com/Lenmoncc/PIC16F877A-Experiments>

（1）整体工程

如图所示，为整体工程目录。此次任务将工程代码模块化，便于后续课设的继续开发。其中，“beep”蜂鸣器模块用来判断是否进入接收中断；“LCD”液晶显示模块用来显示陀螺仪的三轴角度；“JY62”陀螺仪模块为此次的核心代码；“main”为主程序。



（2）主程序部分

#include<pic.h>

#include"LCD.h"

#include"KEY.h"

#include "JY62.h"

#include"LED.h"

#include"beep.h"

\_\_CONFIG(0xFF29);

void main()

{

 Beep\_Init();

 LCD\_Init();

 LCD\_ShowString("1001", 0, 0);

 LCD\_ShowString("Hello World!\n", 1, 0);

 LCD\_Clear();

 KEY\_Init();

 Uart\_Init();

 LCD\_Init();

 while(1) {

*//KEY\_ShowTest();*

  JY62\_ShowData(Roll,Pitch,Yaw);

*//DELAY();*

 }

}

（3）核心代码部分

#include "JY62.h"

#include"LCD.h"

#include"beep.h"

static uint8\_t RxBuffer[11];

static volatile uint8\_t RxState = 0;

static uint8\_t RxIndex = 0;

float Roll,Pitch,Yaw;

void Uart\_Init()

{

    TRISC = 0xC0;

    TXSTA=0X24;

    SYNC=0;

    RCSTA=0X90;

    SPEN=1;

    SPBRG=25; *//9600bps*

    CREN=1;

    GIE=1;PEIE=1;RCIE=1;

}

void interrupt uart\_service()

{

*//RE0 = !RE0;*

    if (RCIF) {

*//使用蜂鸣器判断是否进入中断*

   RE0 = !RE0;

*// DELAY();*

        uint8\_t RxData;

        RxData = RCREG;

        JY62\_ReceiveData(RxData);

    }

}

void JY62\_ReceiveData(uint8\_t RxData)

{

 uint8\_t i,sum=0;

 if (RxState == 0)

 {

  if (RxData == 0x55)

  {

   RxBuffer[RxIndex] = RxData;

   RxState = 1;

   RxIndex = 1;

  }

 }

 else if (RxState == 1)

 {

  if (RxData == 0x53)

  {

   RxBuffer[RxIndex] = RxData;

   RxState = 2;

   RxIndex = 2;

  }

 }

 else if (RxState == 2)

 {

  RxBuffer[RxIndex++] = RxData;

  if(RxIndex == 11)

  {

   for(i=0;i<10;i++)

   {

    sum = sum + RxBuffer[i];

   }

   if(sum == RxBuffer[10])

   {

*//??[-180, 180]*

    Roll = ((int16\_t) ((int16\_t) RxBuffer[3] << 8 | (int16\_t) RxBuffer[2])) / 32768.0 \* 180;

    Pitch = ((int16\_t) ((int16\_t) RxBuffer[5] << 8 | (int16\_t) RxBuffer[4])) / 32768.0 \* 180;

    Yaw = ((int16\_t) ((int16\_t) RxBuffer[7] << 8 | (int16\_t) RxBuffer[6])) / 32768.0 \* 180;

   }

   RxState = 0;

   RxIndex = 0;

  }

 }

}

float JY62\_InfiniteYaw(float yaw)

{

*// static uint8\_t flag=0;*

*// static float last\_yaw = 0;*

*// if (yaw - last\_yaw < -180) {*

*//  flag++;*

*// } else if (yaw - last\_yaw > 180) {*

*//  flag--;*

*// }*

*// last\_yaw = yaw;*

*// return yaw+ flag \* 360.0;*

 float new\_yaw;

 new\_yaw = 180 - (180 - (int)yaw) % 360;

 return new\_yaw;

}

*//Roll,Pitch,Yaw*

void JY62\_ShowData(float roll, float pitch, float yaw)

{

    unsigned char str[11];

*//LCD\_Init();*

    LCD\_ShowString("Pitch:", 0, 0);

    LCD\_ShowString("Yaw:", 1, 0);

 uint16\_t infinite\_yaw = JY62\_InfiniteYaw(yaw);

 sprintf(str, "%3d",infinite\_yaw);

    LCD\_ShowString(str, 1, 4);

    sprintf(str, "%3d",(uint16\_t)pitch);

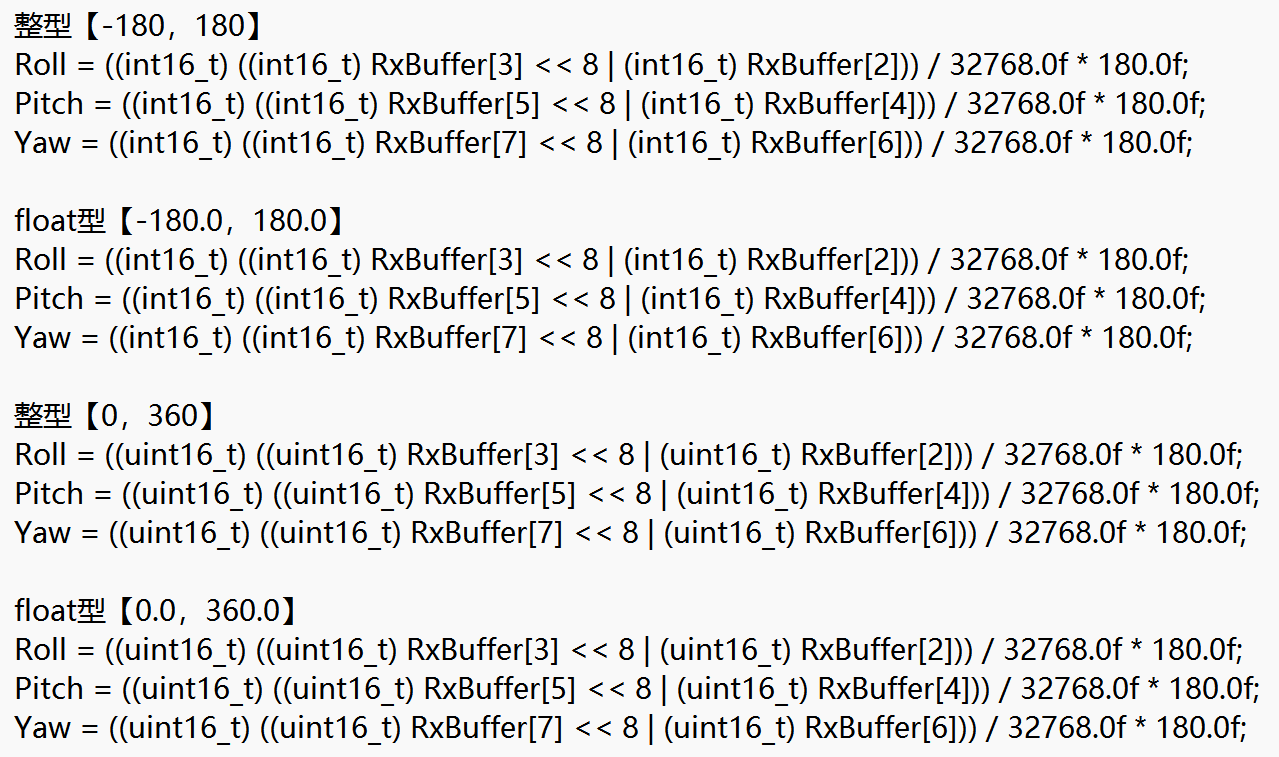
    LCD\_ShowString(str, 0, 6);

*//DELAY();*

*//LCD\_Clear();*

}

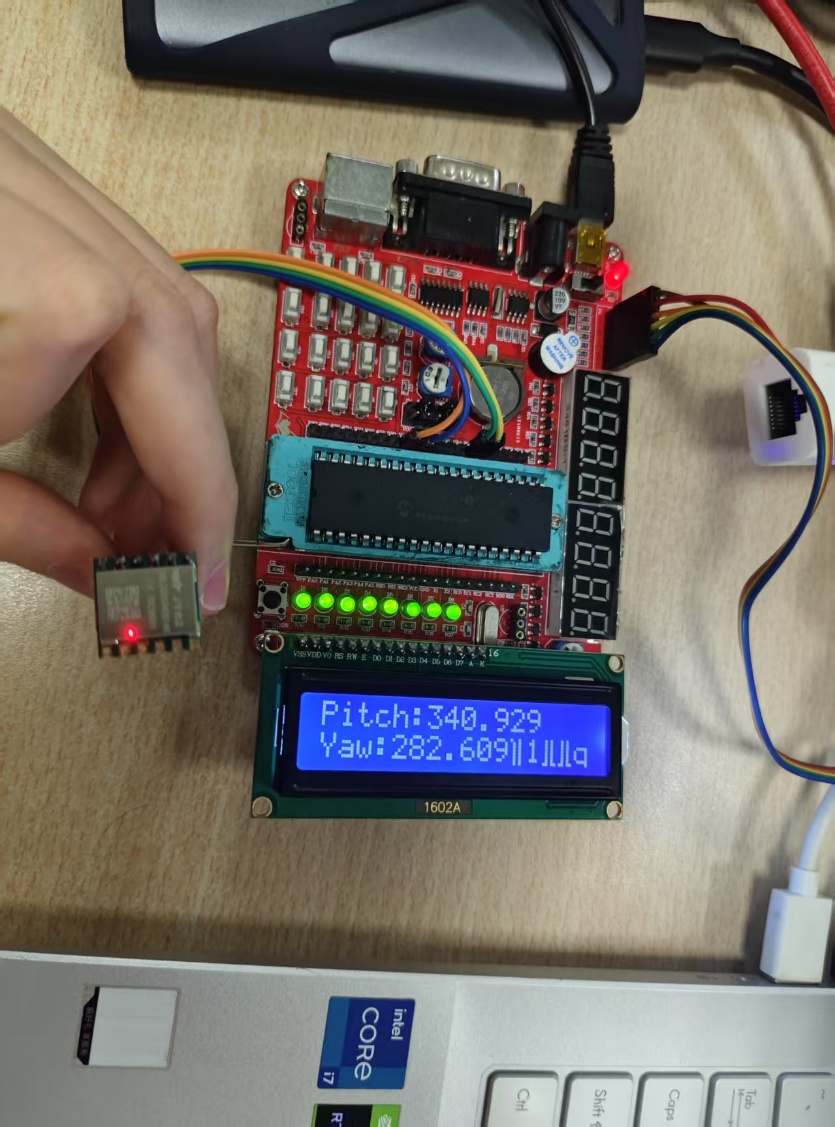
（4）输出范围以及类型的选择



四、实验现象

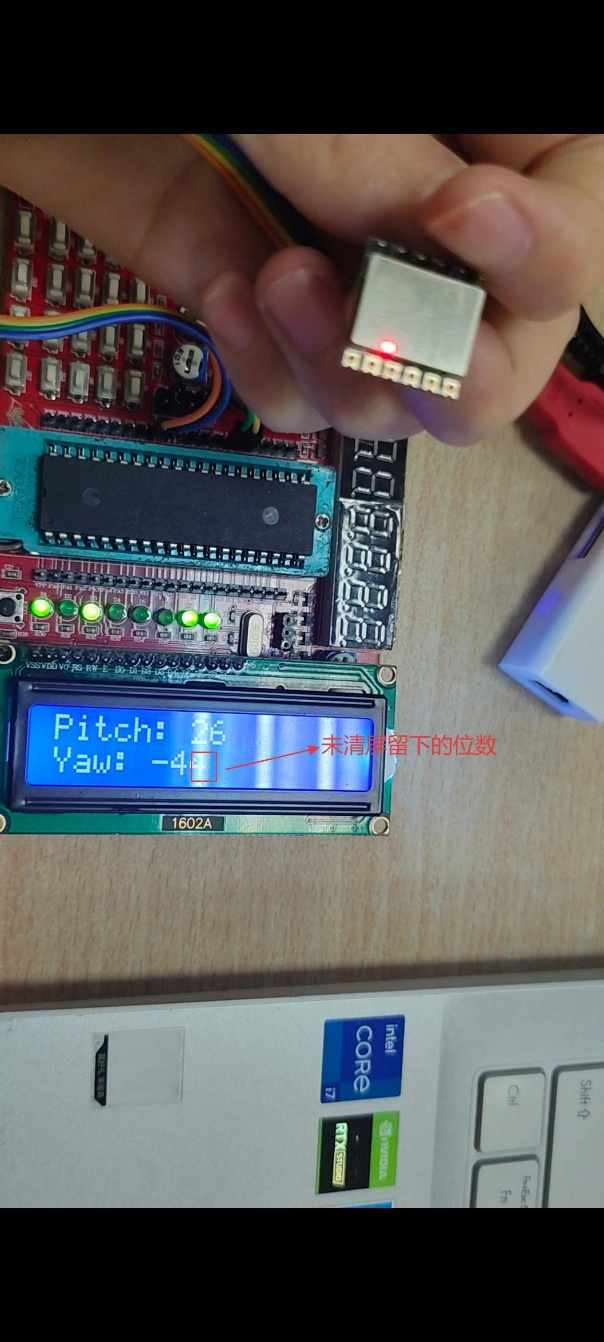
（1）【0，360】的float型显示

由于PIC16F877A单片机为8位的，处理float型数据会比32位的单片机慢，同时LCD的刷新速率也慢。在此过程中，也会频繁的出现卡顿。最终使用uint16\_t数据类型，同时降低LCD的延时以提高其刷新速率。如图所示，为float型卡顿照片。



（2）【-180，180】uint16\_t类型的显示

在使用uint16\_t类型输出时，如果不清屏液晶的话，会造成肉眼数据的误判，即当从三位数变化到两位数时，仍然会看到三个数字，但只有两位数字在变化。其他情况同理。因此，使用不断初始化、显示、清屏的方法避免数据的误判。左图为使用清屏，右图为未使用清屏。

五、实验总结及心得

（1）实验总结

本次实验围绕 JY62 陀螺仪模块的驱动与数据显示展开，通过串口通信、状态机数据解析及 LCD 显示等技术环节，实现了三轴角度信号的实时采集与可视化。

1. 技术实现与关键要点
2. 串口通信配置：依据 JY62 模块参数，选用 9600 波特率（SPBRG=25）进行 UART 通信配置，通过TXSTA、RCSTA寄存器完成异步通信参数设定（8 位数据位、1 位停止位、无校验位），并启用中断机制（RCIE=1）实现数据实时接收，确保波特率误差控制在 3% 以内，避免通信失败。
3. 状态机数据解析：采用三状态机（RxState=0/1/2）识别数据帧头（0x55+0x53），通过累加校验和（SUM）验证数据完整性。当接收满 11 字节数据帧后，通过位运算合并高低字节（如(RxBuffer[3] << 8 | RxBuffer[2])），并按公式(value / 32768.0 \* 180)转换为角度值（°），实现从十六进制原始数据到物理量的映射。
4. 角度突变处理：针对偏航角（Yaw）在 ±180° 处的跳变问题，采用公式new\_yaw = 180 - (180 - yaw) % 360进行归一化处理，通过偏移 - 模运算 - 反向偏移的数学变换，将角度范围限定在 [-180°, 180°]，避免数值突变导致的显示异常。
5. 数据类型与显示优化：

* 浮点型（float）：初始尝试直接显示浮点型角度（如Pitch:340.929），但受限于 8 位单片机（PIC16F877A）的浮点运算性能，出现 LCD 刷新卡顿现象。
* 无符号整型（uint16\_t）：改用整数类型显示，通过强制类型转换（如(uint16\_t)pitch）结合格式控制符%3d实现固定宽度输出。同时发现不清屏会导致残影误判，最终通过 “初始化 - 显示 - 清屏” 循环提升显示清晰度。

1. 功能验证与结果
2. 成功实现 JY62 模块与单片机的通信，实时解析并显示三轴角度（Roll、Pitch、Yaw），支持 [-180°, 180°] 整型与 [0°, 360°] 浮点型两种范围选择。
3. LCD 显示效果优化后，数据更新流畅度提升，消除了因缓存残留导致的显示错位问题，满足实时监控需求

（2）心得

本次实验成功驱动 JY62 模块并实现数据 LCD 显示，为后续课设奠定基础。过程中深刻体会到串口通信波特率精准配置的关键，需通过公式严格计算寄存器值以控制误差；状态机解析数据帧时，需精准匹配包头与校验和，强化了边界条件处理意识。数据类型选择上，浮点型显示因单片机性能受限出现卡顿，改用 uint16\_t 整型后需解决清屏不及时导致的残影误判问题，最终通过循环刷新优化显示效果。尽管任务完成，仍存在诸多可优化之处，如实际应用中陀螺仪零点偏移的校准、滤波算法提升数据稳定性，以及 LCD 显示效率的进一步优化等。这些实践中的思考与不足，为后续嵌入式项目开发积累了宝贵经验，也明确了技术深化的方向