



课 程 \_\_\_\_传感与检测技术\_\_\_\_\_\_

项 目 基于CC2530实现GPS模块u-blox M8N的数据采集

专 业 \_\_\_\_\_\_交通运输工程\_\_\_\_\_\_\_\_\_\_

班 级 \_\_\_\_\_\_交通运输工程2301\_\_\_\_\_\_\_\_\_

学 号 \_\_\_\_\_2231801015\_\_\_\_\_\_\_\_\_

姓 名 \_\_\_\_\_\_\_\_翁泽康\_\_\_\_\_\_\_\_\_\_\_\_

日 期 \_\_\_\_\_\_2023.12.25\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| 实  验  任  务 | 基于CC2530实现GPS模块u-blox M8N的数据采集 |
| 实  验  报  告  内  容 | 1. 关键技术和传感器介绍：  1.1 CC2530： 由德州仪器（Texas Instruments）研发的CC2530是一款低功耗无线系统芯片。它基于IEEE 802.15.4标准，内置2.4GHz无线收发功能，适用于低功耗无线通信和物联网（IoT）应用。  1.2 u-blox M8N：u-blox M8N是一款高性能、高精度的GPS模块，通过UART（串口通信）接口与主控器件进行数据交换。该模块支持多个卫星系统，具备快速冷启动、高灵敏度和低功耗特性。无论是车载导航、航空航天还是物联网应用，u-blox M8N都能提供可靠的定位性能和多接口选择。  1.3 UART通信：UART（通用异步收发传输）是一种简单可靠的串行通信接口，通过发送和接收线实现设备之间的双向数据传输。采用异步传输方式，通过帧结构进行数据传输，并支持设定波特率以调整传输速率。广泛应用于嵌入式系统和通信设备，适用于短距离的数据传输，常见于传感器数据采集、串口调试和外设控制等应用  1.4 NMEA协议：\*\* NMEA协议是一种常用的文本协议，用于在导航设备和接收器之间传输位置、速度、时间和其他相关信息。该协议定义了一系列标准化的数据语句格式，提供可读性强且易于解析的位置数据。在航海、航空和车辆导航等领域广泛应用，允许不同设备之间的数据交换和互操作性，实现准确的位置定位和导航功能。  2. 实验目的：  本设计旨在通过UART通信方式，使CC2530与u-blox M8N进行通信。u-blox M8N通过接收来自卫星系统（如GPS、GLONASS、BeiDou、Galileo等）的无线信号，并利用内部专用的芯片和算法对信号进行处理和解码，从而获取位置、速度和时间等相关信息。CC2530通过程序（涉及NMEA协议）读取经度和纬度的数值，最后通过串口与电脑通信，通过上位机（电脑）实时监测CC2530读取经纬度的数值。  3.1 操作步骤：  1. 通过芯片手册了解传感器的读取方式和通讯方式。  2. 利用IAR进行程序编写，并生成hex文件。  3. 使用烧入线将电脑与CC2530相连，通过烧录软件（如SmartRF Flash Programmer）将程序烧写进CC2530。  4. 将u-blox M8N与CC2530进行UART通信，并通过USB转TTL线将电脑与CC2530连接。  5. 通过上位机—串口调试助手(XCOM)观察u-blox M8N读取的经度和纬度。  3.2读取u-blox M8N数值的步骤：  3.2.1流程图：    CC2530首先进行UART初始化，然后通过`gpsDataCollection()`函数读取u-blox M8N的原始数据。在这个函数中，它持续从UART接口接收数据，检查NMEA协议的起始和结束标记，并将接收到的数据存储在一个数组中。一旦接收到完整的NMEA协议，它会调用`processNmeaData()`函数，从NMEA数据中提取纬度和经度的信息。处理完NMEA数据后，最终通过上位机显示经度和纬度的数据。  3.2.2关键程序：  // 1. 配置UART通信引脚  #define UART\_RX\_PORT P0\_2  #define UART\_TX\_PORT P0\_3  #define UART\_RX\_PIN BV(2)  #define UART\_TX\_PIN BV(3)  // 2. 定义全局变量  double latitude = 0.0;  double longitude = 0.0;  // 3. 实现gpsDataCollection()函数  void gpsDataCollection()  {  uint8\_t data;  uint8\_t gpsData[128];  uint8\_t dataIndex = 0;  uint8\_t nmeaStartIndex = 0;  uint8\_t nmeaEndIndex = 0;  while (1) {  // 接收一个字节  data = uartReceiveByte();  // 判断是否是NMEA协议的起始字节  if (data == '$') {  nmeaStartIndex = dataIndex; // 记录NMEA协议的起始位置  gpsData[dataIndex++] = data; // 存储接收到的数据  }  // 判断是否接收到完整的NMEA协议  if (data == '\n') {  nmeaEndIndex = dataIndex; // 记录NMEA协议的结束位置  // 处理接收到的完整NMEA协议数据  uint8\_t nmeaLength = nmeaEndIndex - nmeaStartIndex;  processNmeaData(&gpsData[nmeaStartIndex], nmeaLength);  // 重置索引，准备接收下一条数据  dataIndex = 0;  nmeaStartIndex = 0;  nmeaEndIndex = 0;  }  }  }  // 4. 实现processNmeaData()函数  void processNmeaData(uint8\_t \*nmeaData, uint8\_t dataLength)  {  // 检查数据是否以"$"开头，以保证是有效的NMEA协议数据  if (nmeaData[0] == '$') {  // 检查数据是否是GGA协议，GGA协议中包含经纬度信息  if (strncmp((const char\*)nmeaData + 1, "GGA", 3) == 0) {  // 使用strtok函数分割数据字段  char \*token = strtok((char\*)nmeaData, ",");  uint8\_t fieldIndex = 0;  while (token != NULL) {  // 经纬度信息通常在第3个和第5个字段中  if (fieldIndex == 2) {  // 解析纬度信息（以度为单位）  latitude = atof(token);  // 在此处进行纬度信息的处理  } else if (fieldIndex == 4) {  // 解析经度信息（以度为单位）  longitude = atof(token);  // 在此处进行经度信息的处理  }  token = strtok(NULL, ",");  fieldIndex++;  }  }  }  }  // 5. 实现main()函数  int main(void)  {  uartInit(); // 初始化UART  while (1) {  gpsDataCollection(); // 开始GPS数据采集  // 在这里添加打印经纬度的代码  printf("Latitude: %.6f\n", latitude);  printf("Longitude: %.6f\n", longitude);  }  return 0;  } |
| 实  验  报  告  内  容 | 4结论：  通过在串口调试助手(XCOM)上对数据结果的观察，本实验能准确的通过CC2530芯片读取u-blox M8N从卫星接收到的经度和纬度的数据。窗体顶端  IMG_256  进一步分析这些数据，我们可以利用经度和纬度信息来确定具体的地理位置、进行导航、追踪移动物体等。这对于许多应用领域都非常有用，例如车辆定位、航海导航、户外运动等。  **思考：如果我们把接收到的位置信息直接用于车辆导航或无人机的飞行导航会有什么问题？可以用什么方法解决这些问题？**  问题：  1.精度问题：u-blox M8N模块的精度一般在几米范围内，但这可能不足以满足高精度导航需求，特别是在城市环境中或需要精确避障的无人机飞行中。  2. 更新率限制：GPS数据的更新率可能不足以应对快速变化的环境。在高速移动的情况下，如车辆或无人机快速行驶，数据更新率可能不足以实时反映位置变化。  3. 信号遮挡：在城市峡谷、室内或树木覆盖区域，GPS信号可能会受到遮挡，导致定位不准确或丢失信号。  4. 多路径效应：在城市环境中，GPS信号可能会在建筑物之间反射，造成多路径效应，进一步影响定位精度。  5. 动态环境适应性：用于车辆或无人机的导航系统需要能够适应动态变化的环境，例如避开突然出现的障碍物。  解决方法：  增加辅助传感器：使用惯性测量单元(IMU)、激光雷达(LiDAR)或视觉摄像头等辅助传感器，可以提高导航系统的整体精度和适应性。  差分GPS（DGPS）：通过使用地面基站提供的校正信号，差分GPS可以显著提高定位精度。  实时运动融合算法：结合GPS数据和其他传感器数据，使用先进的数据融合算法可以提高导航系统的精度和稳定性。  信号增强技术：使用更高增益的天线或信号增强技术可以在信号较弱的环境中提高GPS的性能。  地图数据融合：结合详细的地图数据可以帮助系统更好地理解环境，特别是在城市区域。 |
| 教  师  评  定 |  |