بسم الله الرحمن الرحیم

گزارش کار پروژه درس سیستم های ریزپردازنده ای و مدارهای واسطه

زهرا عربی ۹۵۲۳۰۸۳

سپیده زهدی ۹۵۲۳۰۵۵

ریحانه سادات سجادی ۹۵۲۳۰۵۹

۱۳۹۸-۱۳۹۷

- NRF24_begin :

ورودی های این تابع نام GPIO ، نام SPI و پین های CE (Chip Enable) و CSN (Chip Select Not) می باشد.

CE (Chip Enable) : در هنگام فرستادن یا دریافت دیتا، باید بیشتر از 10µs (طبق بند ۱۱ صفحه ۶۸ دیتاشیت) این پین فعال شود (CE=1).

CSN (Chip Select Not) : در هنگام نوشتن و خواندن با SPI باید این پین فعال شود (CSN=0).

برای کپی کردن تنظیمات SPI در یک Structure جهت استفاده در کتابخانه، از تابع memcpy استفاده کردیم. این تابع در کتابخانه stdio.h تعریف شده است و به صورت زیر کار می کند.

Memcpy :

دارای سه ورودی می باشد (ورودی اول، ورودی دوم، سایز ورودی دوم). ورودی دوم را در ورودی اول کپی می کند.

همان طور که گفته شد در حالت ایده آل ← CE=0 و CSN=1

طبق جدول شماره ۱۳ در دیتاشیت، زمانی که از کریستال داخلی برای تولید کلاک استفاده می کنیم، جهت رفتن از حالت Powerdown به حالت standby باید حداکثر 1.5ms تاخیر داشته باشیم.

سپس تمام رجیستر های NRF24 را طبق جدول شماره ۲۴ در دیتاشیت تنظیم می کنیم. (حالت پیش فرض) برای این کار از توابع NRF24_write_register و NRF24_write_registerN استفاده می کنیم.

- NRF24_write_register :

این تابع دو ورودی دارد (آدرس ریجیستر و دیتا).

همان طور که گفته شد ابتدا CSN=0 و در انتها نیز CSN=1. طبق جدول شماره ۱۶ در دیتاشیت زمانی که بخواهیم از طریق SPI بنویسیم، باید بیت ششم یک باشد. در نتیجه آدرس ورودی را با 0x20، or می کنیم. این تابع یک آرایه دوتایی ایجاد کرده و در خانه اول آن، آدرس و در خانه دوم دیتا را می ریزد. سپس این آرایه را از طریق SPI به NRF24 پاس می دهد.

- NRF24_write_registerN :

این تابع سه ورودی دارد (آدرس رجیستر و دیتا و سایز دیتا).

مانند تابع NRF24_write_register عمل می کند با این تفاوت که دیتای آن می تواند اندازه ای بزرگتر از یک بایت داشته باشد. به همین دلیل ساز دیتا را نیز باید به عنوان ورودی بدهیم.

- NRF24_ACTIVATE_cmd :

همان طور که در پاراگراف c صفحه ۵۸ دیتاشیت گفته شده، برای فعال کردن باید ابتدا دستور Activate SPI را که طبق جدول شماره ۱۶ برابر با 0x50 است را فرستاده و سپس مقدار 0x73 را بفرستیم که این تابع این دو مقدار را در یک آرایه دوتایی ریخته و سپس آن را ارسال می کند.

- printRadioSettings :

تنظیمات صورت گرفته شده را چاپ کرده و به کاربر نمایش می دهد. به این صورت که آدرس رجیستر ها را به توابع NRF24_read_register و NRF24_read_registerN پاس داده و دیتاها را دریافت می کند. سپس آن ها را به وسیله UART نمایش می دهد.

- NRF24_read_register :

این تابع یک ورودی دارد (آدرس ریجیستر).

همان طور که گفته شد ابتدا CSN=0 و در انتها نیز CSN=1. طبق جدول شماره ۱۶ در دیتاشیت زمانی که بخواهیم از طریق SPI بخوانیم، باید آدرس ورودی را با 0x1F، and کنیم. این تابع یک آرایه دوتایی ایجاد کرده و در خانه اول آن، آدرس رجیستر را می ریزد. سپس این آرایه را از طریق SPI به NRF24 پاس می دهد. سپس دیتایی را که از طریق SPI دریافت کرده را در خانه دوم می ریزد. سپس همین دیتا را در خروجی می ریزد.

- NRF24_read_registerN :

این تابع سه ورودی دارد (آدرس رجیستر، آرایه و اندازه آرایه).

این تابع همانند تابع NRF24_read_register عمل کرده با این تفاوت که سایز دیتای دریافتی بیشتر از یک بایت می تواند باشد.

- NRF24_initialize :

جهت اعمال تغییرات مورد نظر استفاده می شود.

- NRF24_setRetries :

این تابع تنظیم می کند که آرایه ارسالی چند بار و با چه مدت زمان تاخیری ارسال شود.

طبق جدول ۲۴ زمان تاخیر با چهار بیت باارزش و تعداد ارسال ها با چهار بیت کم ارزش SETUP_RETR مشخص می شود.

- NRF24_setAutoAck :

ما در این پروژه از حالت Auto Acknowledge استفاده نمی کنیم اما به دلیل آنکه به صورت رندوم تغییر نکند، این تابع را قرار داده و تمام بیت های EN_AA را صفر می کنیم.

- NRF24_setPALevel : (تنظیم توان خروجی)

برای تنظیم Power به کار می رود. این مقدار می تواند طبق جدول شماره ۲۴، 0 dBm,-6 dBm,-12 dBm,-18 dBm باشد.

- NRF24_setDataRate : (تنظیم سرعت لینک مخابراتی)

برای تنظیم Data Rate به کار می رود. طبق جدول شماره ۲۴ می تواند 1 Mbps یا 2 Mbps باشد.

- NRF24_setCRCLength :

CRC می تواند یک یا دو بایت باشد و کنترل کننده است. این تابع طول آن را تنظیم می کند. طبق جدول شماره ۲۴ این تنظیمات توسط بیت دوم CONFIG صورت می گیرد.

- NRF24_disableDynamicPayloads :

این تابع جهت غیر فعال کردن Dynamic Payload طبق جدول شماره ۲۴ بیت دوم FEATURE را صفر قرار می دهد.

- NRF24_setPayloadSize :

طبق جدول شماره ۲۴، ماکزیمم Payload Size می تواند ۳۲ بایت باشد. در این تابع ما یک Constant به اندازه ۳۲ بایت تعریف کرده و ورودی تابع را با این Constant مقایسه کرده و مینیمم آن ها را در payload_size قرار می دهیم. بدین ترتیب Payload Size از ۳۲ بایت بیشتر نخواهد شد.

- NRF24_setChannel : (تنظیم کانال فرکانسی)

طبق جدول شماره ۲۴، در RF_CH می توانیم Channel را تنظیم کنیم. در این رجیستر بیت هفتم باید صفر باشد و بیت های دیگر با توجه به فرکانس می توانند صفر یا یک باشند. پس ماکزیمم مقدار آن می تواند ۱۲۷ باشد. در این تابع مینیمم مقدار ورودی و Constant تعریف شده (۱۲۷) را در RF_CH می ریزیم.

حال طبق configuration در صفحه ۶۸ دیتاشیت، بعد از تنظیم کردن موارد مورد نظر، باید TX Payload و RX Payload را با 0xFF پر کنیم. طبق جدول شماره ۱۶، آدرس FLUSH_TX و FLUSH_RX ، 0xE1 و 0xE2 می باشد.

- NRF24_powerDown :

طبق جدول شماره ۲۴، با استفاده از بیت اول CONFIG می توانیم این کار را انجام دهیم. بنابراین بیت اول CONFIG را با استفاده از قابلیت and صفر می کنیم.

حال دوباره تنظیمات را با استفاده از تابع printRadioSettings نمایش می دهد.

- nrf24_DebugUART_Init :

با استفاده از تابع memcpy تمام مشخصات Uart را در یک Structure جدید ذخیره می کند.

- NRF24_openReadingPipe :

در این تابع ما شماره Pipe و آدرس را به عنوان ورودی می دهیم. طبق جدول شماره ۲۴، ما شش Pipe داریم که RX_ADDR_P0 و RX_ADDR_P1 ۵ بایتی و بقیه یک بایتی هستند. با استفاده از تابع NRF24_write_register آدرس را در NRF24_ADDR_REGS می ریزد. سپس payload_size تنظیم شده در تابع NRF24_setPayloadSize را در RX_PW_Pipe مورد نظر می ریزد. سپس برای Enable کردن Pipe مورد نظر (با استفاده از جدول ۲۴) بیت گفته شده را در EN_RXADDR یک می کند.

- NRF24_startListening :

این تابع در گیرنده به کار می رود. ابتدا در CONFIG با یک کردن بیت های شماره صفر و یک، آن را Power Up کرده و در حالت PRX قرار می دهد. برای اینکه بتوانیم اطلاعات را دریاف کنیم، Flush را خالی می کنیم. سپس CE=1 قرار می دهیم و 150µs صبر می کنیم. طبق شکل شماره ۱۴ صفحه ۳۹ ابتدا CE=1 شده و بعد از 130µs دریافت می کنیم.

- NRF24_available :

ما از این تابع جهت فهمیدن اینکه در حالت گیرنده رفتیم یا خیر استفاده می کنیم.

در این تابع ابتدا REG_STATUS را در status می ریزد و در واقع به غیر از بیت RX_DR، بقیه بیت ها را صفر کرده و داخل result می ریزد. اگر result=1 باشد، آن را داخل REG_STATUS می ریزد. در این جا بیت TX_DS را نیز چک کرده و در صورتی که یک باشد، آن را صفر می کند.

- NRF24_read :

در این تابع همان طور که گفته شد جهت خواندن متن ارسال شده ابتدا CSN=0 و در انتها CSN=1 قرار می دهیم. طبق جدول شماره ۱۶ جهت خواندن باید R_RX_PAYLOAD را به SPI بفرستیم. سپس بافر دریافت می کنیم.

- NRF24_stopListening :

در این تابع چون نمی خواهیم فعلا کاری انجام دهیم CE=0 . سپس Flush ها را خالی کرده و آن ها را برای ارسال داده آماده می کنیم.

- NRF24_openWritingPipe :

در این تابع ما شماره Pipe و آدرس را به عنوان ورودی می دهیم. طبق جدول شماره ۲۴، ما شش Pipe داریم که RX_ADDR_P0 و RX_ADDR_P1 ۵ بایتی و بقیه یک بایتی هستند. با استفاده از تابع NRF24_write_register آدرس را در NRF24_ADDR_REGS می ریزد. سپس آدرس را در TX_ADDR می ریزیم و اندازه Pipe را نیز در RX_PW_PIPE مورد نظر می ریزیم.

- NRF24_startWrite :

در این تابع ابتدا در REG_CONFIG ، آن را Power up کرده و در حالت TX قرار می دهد. سپس طبق جدول شماره ۱۳، برای رفتن به حالت Standby ، 150µs صبر می کنیم.

سپس همان طور که گفته شد جهت نوشتن متن ارسال شده ابتدا CSN=0 و در انتها CSN=1 قرار می دهیم. طبق جدول شماره ۱۶ جهت نوشتن باید W_TX_PAYLOAD را به SPI بفرستیم. سپس بافر مورد نظر را نیز به SPI می فرستیم. سپس همان طور که گفته شد در هنگام فرستادن دیتا، باید بیشتر از 10µs (طبق بند ۱۱ صفحه ۶۸ دیتاشیت) پین CE فعال شود که ما در 20µs آن را فعال کردیم.

- NRF24_write :

در این تابع ابتدا REG_STATUS را خوانده و در واقع به غیر از بیت TX_DS، بقیه بیت ها را صفر کرده و داخل retstatus می ریزد. اگر retstatus =1 باشد، آن را داخل REG_STATUS می ریزد. در این جا بیت RX_DR را نیز چک کرده و در صورتی که یک باشد، آن را صفر می کند. سپس با استفاده از تابع HAL_GetTick شروع به شمارش می کنیم. تا زمانی که در حالت TX باشد و تعداد ارسال اینتراپت ها به ماکسیمم نرسیده باشد و زمان هم به 10ms نرسیده باشد، REG_STATUS را خوانده و داخل status می ریزد. بعد از این که یکی از این شرط ها نقض شد، بیت TX_DS را clear کرده و TX را Flush می کند.

- NRF24_DelayMicroSeconds :

برای ایجاد تاخیر زمانی میکروثانیه از تابعی در اینترنت استفاده کردیم.

- NRF24_csn و NRF24_ce :

عمل Write Pin را انجام می دهند.

سوال ۳:

با توجه به اینکه ماکزیمم DataRate برابر با 2Mbps و ماکزیمم UART برابر با 4.5Mbps می باشد. در نتیجه ما نمی توانیم بیشتر از 2Mbps تنظیم کنیم.