

EmedRF Firmware Version 2.7
EmbedRF LLC
www.embedrf.com

Overview

The EmbedRF LLC version 2.7 firmware performs several major functions including the following: (1) serial UART for configuration of the system, (2) EEPROM for storing system settings in non-volatile memory, (3) CC1101 transceiver configuration, and (4) timing system which controls the wake-up and sleep functionality of the system. Please see the EmbedRF Datasheet for the full functionality of the EmbedRF module.

Source Files

The EmbedRF firmware consists of two files:

Conrol.c
Control.h

Compilation Information

Compiled with Hitech PICC compiler version 9.5, MPLA IDE v8.40. Memory map is given below:

Memory Usage Map:

Program space:

CODE	used	F4Bh (3915)	of	1000h words	(95.6%)
CONST	used	90h (144)	of	1000h words	(3.5%)
ENTRY	used	16h (22)	of	1000h words	(0.5%)
STRING	used	0h (0)	of	1000h words	(0.0%)

Data space:

BANK0	used	52h (82)	of	60h bytes	(85.4%)
BANK1	used	3Fh (63)	of	50h bytes	(78.8%)
BANK2	used	29h (41)	of	50h bytes	(51.3%)
COMBANK	used	1h (1)	of	10h bytes	(6.3%)

EEPROM space:

EEDATA	used	0h (0)	of	100h bytes	(0.0%)
--------	------	---------	----	------------	---------

ID Location space:

IDLOC	used	0h (0)	of	4h bytes	(0.0%)
-------	------	---------	----	----------	---------

Configuration bits:

CONFIG	used	1h (1)	of	1h word	(100.0%)
--------	------	---------	----	---------	----------

Summary:

Program space	used	FF1h (4081)	of	1000h words	(99.6%)
Data space	used	BBh (187)	of	100h bytes	(73.0%)

EEPROM space used 0h (0) of 100h bytes (0.0%)
 ID Location space used 0h (0) of 4h bytes (0.0%)
 Configuration bits used 1h (1) of 1h word (100.0%)

Function Descriptions

The function descriptions are included in the table below.

Function Name	Description
Main	The main entry point for the firmware. It begins by initializing the hardware, the ports, the SPI interface, and the transceiver. The main loop is in this function. During the main loop, the program waits for (1) a timer expiration (indicating it is time to transmit / receive data, (2) a command from the serial UART.
int_srvc	Interrupt routine. All interrupts are checked and serviced here.
CtrlGetAD	This function is used to obtain an A/D reading. The channel indicates which A/D is recorded.
CtrlSaveIdGen	This function saves the transmit/receive ID of the transceiver into the EEPROM
CtrlSetMicroOff	Turns off wireless data transmission
CtrlSetMicroOn	Turns on wireless data transmission
CtrlConfigAnalog	Configures the A/D converter
IncrementId	Increments ID by 1. This is used to periodically increment either the transmit ID or the receive ID for low-power multi-point to point operation
CtrlSendRfPacket	Transmits a wireless data packet, and based on the input value, either waits for a data packet to receive, and/or performs a calibration.
CtrlPutDataOnUart	Sends data received by the transceiver out the UART
CtrlPutId	Sends 3-byte ID out the UART
CtrlPutAcquisitionMode	Sends the data acquisition mode out the UART
CtrlPutChars	Sends two input characters out the UART
CtrlVerifyUart	Checks to see if the input command has the expected number of bytes. The expected number of bytes for a given command is stored within the array called ReceiveLength
SetChannel	Sets the frequency of the RF channel used for transmission / reception. NOTE: You must use an antenna and matching circuit that is appropriate for the RF channel selected. The off-the-shelf EmbedRF modules come with a 915 MHz antenna configuration
CtrlProcessUart	This function is called from the main loop when a UART command has arrived. This function processes the command and performs the required functionality of the command.

CtrlGetPower	Retrieves the power configuration (e.g. 0 dbm) of the device from the EEPROM memory.
CtrlGetNetMode	Retrieves the network configuration of the device from the EEPROM.
CtrlGetTxMode	Retrieves the wireless transmission configuration of the device from the EEPROM
CtrlSetDigital	Sets the two digital I/O channels to outputs and to either a 1 or a zero.
CtrlSetAcquisitionMode	Sets the analog / digital operating modes for the 4 I/O channels
CtrlGetAcquisitionMode	Retrieves the analog / digital operating modes from the EEPROM.
CtrlSetTxMode	Sets the wireless transmission mode of the device and stores the setting in EEPROM.
CtrlSetTransmissionInterval	Sets the period of time over which the device transmits / receives data (i.e. 0.25-12.75 seconds). Stores the interval within the EEPROM.
CtrlInitializeStates	This function is called at boot-up only. It retrieves all of the previous or default settings from the EEPROM
CtrlSetPower	Sets the power configuration (e.g. 0 dbm) of the device and stores within the EEPROM.
CtrlAdjustOsc	This function uses the external 32768 crystal to tune the internal oscillator of the microcontroller such that we can ensure that all embedRF devices are running at the same frequency within some acceptable tolerance
SpiStrobe	Function for writing a strobe command to the CC1101
RfPowerDown	Powers down the CC1101 into low-power mode
CtrlWaitForRfPacket	Waits for an RF packet to arrive, and then based on the input parameters, transmits a data packet and/or performs a calibration
SpiReadStatus	Reads a status byte from the CC1101
Wait	Waits an period of time dependent on the input parameter.
ChipSelect	Performs a chip select, necessary for SPI communication
PowerUpResetCC1100	Resets the CC1101
SpiWriteReg	Writes a byte to a given address in the CC1101
SpiWriteBurstReg	Writes an array of bytes to a segment of address locations in the CC1101
SpiReadBurstReg	Reads an array of bytes from an address location in CC1101
RfWriteRfSettings	Configures the CC1101 by writing a set of configuration parameters to the CC1101 memory
UartInit	Initializes the UART and sets the baud rate to either 1200, 2400, 9600 or 19200
UartPutChar	Puts a single character out the UART
UartClearBuffer	Sets all UART buffer values to zero
UartAddCharToBuffer	Adds a character to the UART-received buffer. Also

	verifies that if it is the first byte received, that it is a valid command byte (i.e. it is present in the ReceiveVals array)
EEReadEEPROM	Reads a byte from the EEPROM at a given address location.
inittmr	This function initializes the two timers used by the device.
init	This function initializes the oscillator, interrupts, timers, SPI, and other device parameters.
initpio	Initializes the ports on the device
initspi	Initializes the SPI interface of the device
Sleepmode	Puts the device to sleep. When the device is awakened, it reinitializes the ports. Ensures that the digital I/O ports are maintained during sleep.
TmrInitTimer	Initializes the timer which is used to control the periodic data transmission / reception
divide	Performs division of a long number
TmrGetTickCount	Retrieves the psiTickCount, which keeps track of the wireless data transmission interval.
TmrSetTransmissionInterval	Sets the psiTickCount, which keeps track of the wireless data transmission interval
TrmAdjustTimer	Adjusts the expiration time of timer 1 by an amount passed into the function. This adjustment is necessary because the expiration of this timer is once every 2 seconds, which doesn't give the accuracy required for transmission intervals less than 2 seconds.
TmrSetHstOn	Turns the high speed timer on and sets the expiration time.
TmrSetHstOff	Turns off the high speed timer.
TmrHstIncTickCount	Increments the high-speed timer tick count
TmrCheckTimerExpired	Determines if the high-speed timer has expired.
TmrUpdateTimers	Checks to see if any of the timers have expired.