# Board Netlist

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MSP430** | | **NET** | **CC2500** | | **H1** | **TP** |
| P4.7/TBCLK | 22 | NC |  |  |  |  |
| P4.6/TBOUTH/OA1I3 | 21 | P4.6 |  |  |  | 11 |
| P4.5/TB2/A14/OA0I3 | 20 | P4.5 |  |  |  | 10 |
| P4.4/TB1/A13/OA1O | 19 | P4.4 |  |  |  | 9 |
| P4.3/TB0/A12/OA0O | 18 | P4.3 |  |  |  | 8 |
| P4.2/TB2 | 17 | NC |  |  |  |  |
| P4.1/TB1 | 16 | NC |  |  |  |  |
| P4.0/TB0 | 15 | NC |  |  |  |  |
| P3.7/A7/OA1I2 | 26 | NC |  |  |  |  |
| P3.6/A6/OA0I2 | 25 | NC |  |  |  |  |
| P3.5/UCA0RXD/UCA0SOMI | 24 | RXDO |  |  | 1 |  |
| P3.4/UCA0TXD/UCA0SIMO | 23 | TXDO |  |  | 6 |  |
| P3.3/UCB0CLK/UCA0STE | 12 | P3.3\_UCB0CLK | SCLK | 1 |  | 16 |
| P3.2/UCB0SOMI/UCB0SCL | 11 | P3.3\_UCB0SOMI | SO(GDO1) | 2 |  | 15 |
| P3.1/UCB0SIMO/UCB0SDA | 10 | P3.1\_UCB0SIMO | SI | 20 |  | 18 |
| P3.0/UCB0STE/UCA0CLK/A5 | 9 | P3.0\_UCB0STE | CSN | 7 |  | 17 |
| P2.7/XOUT | 2 | GDO2 | GDO2 | 3 |  | 14 |
| P2.6/XIN | 3 | GDO0 | GDO0 (ATEST) | 6 |  | 13 |
| P2.5/Rosc | 40 | NC |  |  |  |  |
| P2.4/TA2/A4/VREF+/VeREF+/OA1O0 | 28 | P2.4 |  |  |  | 7 |
| P2.3/TA1/A3/VREF-/VeREF-/OA1I1/OA1O | 27 | P2.3 |  |  |  | 6 |
| P2.2/TA2/A2/OA0I1 | 8 | P2.2 |  |  |  | 5 |
| P2.1/TAINCLJ/SMCLK/A1/OA0O | 7 | P2.1 |  |  |  | 4 |
| P2.0/ACLK/A0/OA0I0 | 6 | P2.0 |  |  |  | 3 |
| P1.7/TA2/TDO/TDI | 36 | NC |  |  |  |  |
| P1.6/TA1/TDI/TCLK | 35 | NC |  |  |  |  |
| P1.5/TA0/TMS | 34 | NC |  |  |  |  |
| P1.4/SMCLK/TCK | 33 | NC |  |  |  |  |
| P1.3/TA2 | 32 | NC |  |  |  |  |
| P1.2/TA1 | 31 | P1.2 (S1) |  |  |  |  |
| P1.1/TA0 | 30 | P1.1 (LED1) |  |  |  |  |
| P1.0/TACLK/ADC10CLK | 29 | P1.0 (LED0) |  |  |  |  |
| RST/NMI/SBWTDIO | 5 | #RST/SBWTDIO |  |  | 4 |  |
| TEST/SBWTCK | 37 | TEST/SBWTCK |  |  | 3 |  |

# Radio API

mrfi\_spi.c

void mrfiSpiInit(void)

uint8\_t mrfiSpiCmdStrobe(uint8\_t addr)

uint8\_t mrfiSpiReadReg(uint8\_t addr)

void mrfiSpiWriteReg(uint8\_t addr, uint8\_t value)

void mrfiSpiWriteTxFifo(uint8\_t \* pData, uint8\_t len)

void mrfiSpiReadRxFifo(uint8\_t \* pData, uint8\_t len)

mrfi\_radio.c

void MRFI\_Init(void)

uint8\_t MRFI\_Transmit(mrfiPacket\_t \* pPacket, uint8\_t txType)

void MRFI\_Receive(mrfiPacket\_t \* pPacket)

void MRFI\_RxOn(void)

void MRFI\_RxIdle(void)

void MRFI\_Sleep(void)

void MRFI\_WakeUp(void)

void MRFI\_GpioIsr(void)

int8\_t MRFI\_Rssi(void)

uint8\_t MRFI\_RandomByte(void)

void MRFI\_DelayMs(uint16\_t milliseconds)

void MRFI\_ReplyDelay()

void MRFI\_PostKillSem(void)

uint8\_t MRFI\_GetRadioState(void)

mrfi\_f1f2.c

void MRFI\_SetLogicalChannel(uint8\_t chan)

void MRFI\_SetRFPwr(uint8\_t idx) 3 settings (0-2) - Transmit uses 2 hard coded (0dbm)

uint8\_t MRFI\_SetRxAddrFilter(uint8\_t \* pAddr)

void MRFI\_EnableRxAddrFilter(void)

void MRFI\_DisableRxAddrFilter(void);

mrfi.h

void MRFI\_RxCompleteISR(void); /\* populated by code using MRFI \*/

# Radio API Pseudocode

MRFI\_Init()

configure GDO0 pin as input

mrfiSpiInit()

configure csn, sclk, si as output, so as input (via MSP P3 registers)

drive csn high

initialize SPI by configuring UCB0CTL0,1, UCB0BR0,1, P3SEL

send reset command strobe

for each radio configuration register

mrfiSpiWriteReg(write values from mrfiRadioCfg array)

set software state to idle

MRFI\_SetLogicalChannel(0)

turn Rx off

write mfriLogicalChanTable[0] to CHANNR register

if Rx was on turn it back on

MRFI\_SetRFPwr(2)

turn Rx off

write mrfiPowerTable[2] to PA\_TABLE0 register

if Rx was on turn it back on

go to Rx mode

use rssi to populate random seed

turn radio off

set data rate (not sure how it gets to the radio, or if)

clear frame buffers

configure GDO0 as sync signal

configure sync pin interrupt on falling edge

clear sync pin interrupt bit

enable interrupts (board level)

MRFI\_Transmit(packet, txtype)

turn Rx off

write packet to transmit FIFO

if txtype is forced

enable Tx

wait for transmit to complete

clear the interrupt

else cca transmit

while forever

enable Rx

wait for rssi to be valid

clear PA\_PD pin interrupt flag (something to do with sensing a transmission)

initiate transmit

if PA\_PD interrupt flag is set

wait for transmit to complete

break from while

else cca failed

if retries left

do random backoff

else

set return value to fail

flush transmit FIFO

if Rx was on turn it back on

BSP\_ISR\_FUNCTION()

MRFI\_GpioIsr()

Mrfi\_SyncPinRxIsr()

MRFI\_RxCompleteISR()

dispatchFrame()

spCallback -->

sCB() in main\_AP.c

# CC2500 Configuration Register Defaults

|  |  |  |  |
| --- | --- | --- | --- |
| MRFI\_SETTING\_ | IOCFG0 | 6 |  |
| MCSM1 | 0x3c |  |
| MCSM0 | 0x18 | 0x10 | (0x18 & (0x4 | 0x8)) |
| PKTLEN |  | MRFI\_MAX\_FRAME\_SIZE – MRFI\_LENGTH\_FIELD\_SIZE |
| PKTCTRL0 | 0x05 | 0x05 | (0x05 & 0x40) |
| SMARTRF\_ SETTING\_ | FSCTRL1 | 0x07 |  |
| FSCTRL0 | 0x00 |  |
| FREQ2 | 0x5d |  |
| FREQ1 | 0x44 |  |
| FREQ0 | 0xec |  |
| MDMCFG4 | 0x2d |  |
| MDMCFG3 | 0x3b |  |
| MDMCFG2 | 0x73 |  |
| MDMCFG1 | 0x23 | FEC off, min preamble bytes = 4, CHANSPC\_E = 3 |
| MDMCFG0 | 0x3b | CHANSPC\_M = 59 |
| DEVIATN | 0x01 |  |
| FOCCFG | 0x1d |  |
| BSCFG | 0x1c |  |
| AGCCTRL2 | 0xc7 |  |
| AGCCTRL1 | 0x00 |  |
| AGCCTRL0 | 0xb0 |  |
| FREND1 | 0xb6 |  |
| FREND0 | 0x10 |  |
| FSCAL3 | 0xea |  |
| FSCAL2 | 0x0a |  |
| FSCAL1 | 0x00 |  |
| FSCAL0 | 0x11 |  |
| TEST2 | 0x88 |  |
| TEST1 | 0x31 |  |
| TEST0 | 0x0b |  |

|  |  |
| --- | --- |
| mrfiLogicalChanTable | |
| 3 | SMARTRF\_SETTING\_CHANNR |
| 103 |  |
| 202 |  |
| 212 |  |

|  |  |
| --- | --- |
| mrfiPowerTable | |
| 0x46 |  |
| 0x97 |  |
| 0xfe |  |

MRFI\_MAX\_PAYLOAD\_SIZE 20