**RH\_RF95.h**

/// brian.n.norman@gmail.com 9th Nov 2018

/// allows the payload CRC bit to be turned on/off. Normally this should be left on

/// so that packets with a bad CRC are rejected

/// \patam[in] on bool, true turns the payload CRC on, false turns it off

void setPayloadCRC(bool on);

**More Information**

#define RH\_RF95\_REG\_1E\_MODEM\_CONFIG2 0x1e *(00011110)*

#define RH\_RF95\_PAYLOAD\_CRC\_ON 0x04 *(00000100)*

**RH\_RF95.cpp**

void RH\_RF95::setPayloadCRC(bool on)

{

// Payload CRC is bit 2 of register 1E

uint8\_t current = spiRead(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2) & ~RH\_RF95\_PAYLOAD\_CRC\_ON; // mask off the CRC

if (on)

spiWrite(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2, current | RH\_RF95\_PAYLOAD\_CRC\_ON);

else

spiWrite(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2, current);

}

**New program for turning CRC auto clear on and off**

**RH\_RF95.h**

/// Travis McKee 6th Feb 2020

/// allows the payload CRC auto clear bit to be turned on/off. Normally should be left off (Bit 4 = 0)

/// so that packets with a bad CRC are cleared from the FIFO

/// \patam[in] on bool, true turns the payload CRC auto clear off, false turns auto clear off

void setPayloadCRCAutoClear(bool on);

**More Information**

#define RH\_RF95\_REG\_1E\_MODEM\_CONFIG2 0x1e *(00011110) \*already in header file*

#define RH\_RF95\_PAYLOAD\_CRC\_CLEAR\_ON 0x08 *(00001000)*

**RH\_RF95.cpp**

void RH\_RF95::setPayloadCRCAutoClear (bool on)

{

uint8\_t current = spiRead(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2) & ~ RH\_RF95\_PAYLOAD\_CRC\_CLEAR\_ON;

if (on)

spiWrite(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2, current | RH\_RF95\_PAYLOAD\_CRC\_CLEAR\_ON);

else

spiWrite(RH\_RF95\_REG\_1E\_MODEM\_CONFIG2, current);

}

\*The second program does not work as the CRC clear on and off only works on the FSK/OOK mode while we are operating using the LoRa mode