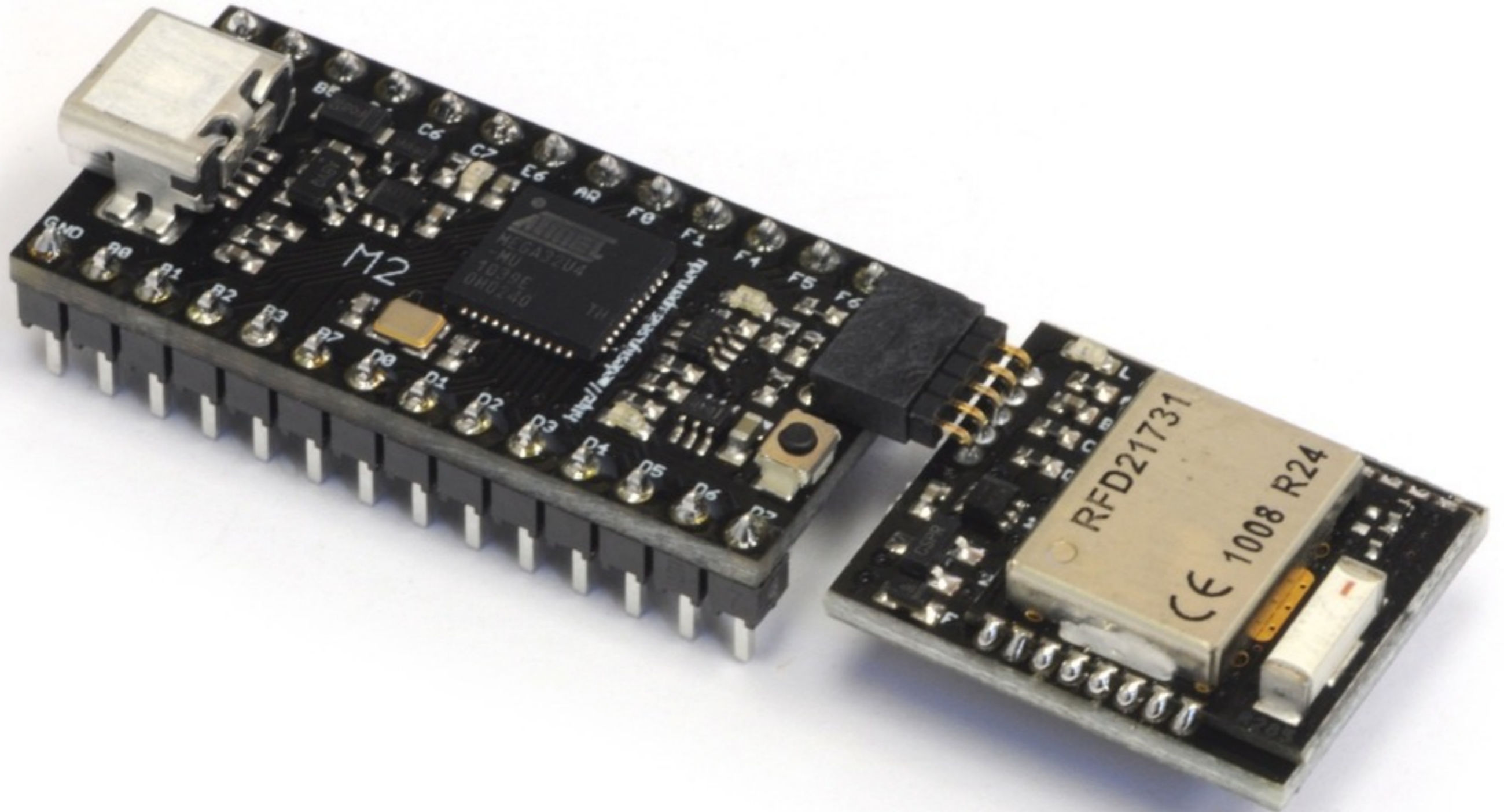


# mRF Wireless Module



# (but first) the mBUS

requires use of pins:

**D0** = CLOCK

**D1** = DATA

**D2** = INTERRUPT

download support files from the wiki:

**m\_bus.h** & **m\_bus.c**



**#include** in main.c



add to your project

to take over pins and configure for use:

**m\_bus\_init();**

write an ISR for INT2

# mRF support

download support files from the wiki:

**m\_rf.h**

&

**m\_rf.c**



**#include** in main.c

add to your project

# mRF functions

**m\_rf\_open**(channel, RXaddress, packet\_length);

listen to a desired **RXaddress** (0x00 to 0xFF)  
over a wireless **channel** (1-32, TX/RX must match)  
with a specified **packet\_length** (1-32, TX/RX must match).

**m\_rf\_read**(buffer, packet\_length);

extract **packet\_length** (1-32, TX/RX must match) bytes  
into a local array **buffer**

**m\_rf\_send**(TXaddress, buffer, packet\_length);

send **packet\_length** (1-32, TX/RX must match) bytes  
stored in a local array **buffer**  
to a desired **TXaddress** (0x00 to 0xFF)

```
#include "m_general.h"
#include "m_bus.h"
#include "m_rf.h"

#define CHANNEL          1
#define RXADDRESS       0x00
#define PACKET_LENGTH   3

char buffer[PACKET_LENGTH] = {0,0,0};

int main(void)
{
    m_clockdivide(0);           // 16 MHz
    m_bus_init();              // enable mBUS
    m_rf_open(CHANNEL,RXADDRESS,PACKET_LENGTH); // configure mRF

    while(1){
        m_rf_send(0xFE,buffer,PACKET_LENGTH);
    }
}

ISR(INT2_vect){
    m_rf_read(buffer,PACKET_LENGTH); // pull the packet
    m_green(TOGGLE);
}
```