

Mchr3k - Arduino

A chronicle of my experiences with Arduino and other electronics.

Sunday, 22 January 2012

Wireless sensor node - manchester lib (3)

Last night I spent several hours trying to improve the RF comms which I am using.

MANCHESTER encoding

The first step was to add indenting and more details comments to MANCHESTER.cpp. I am quite pleased that I spent the time doing this as I now understand how the library works. I will try and describe what I have learnt here for my future reference.

RF comms can be tricky due to a combination of interference and trying to ensure that the TX and RX don't drift out of sync. For example consider a TX which wants to send the number 0 as a 16 bit number. In a naive scheme this would involve TXing 0 for the duration of 16 bits. However, if the receiver is running a bit too fast or slow they could easily read this as 15x 0 or 17x 0. The naive approach is also vulnerable to low frequency interference. The RX could detect bit transitions that were too fast and assume this was interference but there is no limit to how far apart bit transitions can be so low frequency interference will not be detected.

Manchester encoding addresses this by sending each bit as a transition. Zero is HI to LO. One is LO to HI. This means that the midpoint of every sent bit will always contain a transition. The edges of a bit will only contain a transition if the previous bit was the same. For example

- 00 - (HI,LO)(HI,LO)
- 01 - (HI,LO)(LO,HI)

To correctly receive these bits the RX must get in sync with the TX. This ensures that HI,LO,HI,LO is correctly interpreted as 00 instead of the RX missing the initial HI, reading a 1 and then discarding the final LO.

The MANCHESTER library which I am using sends messages as follows.

TX sends

- 14x 0 followed by 1x1 - this is the sync pattern
- 16x data bits
- 2x 0 - I'm not sure why these are sent at the end

RX listens for

- 10x 0 followed by 1x 1
- 16x data bits

The RX listens for a shorter sync sequence because the first few bits are likely to be corrupted. The RX chip uses an amplifier to receive faint signals. This means that when there is no signal the RX will amplify background RF noise until it detects some ones and zeros. When the TX starts sending a message the RX chip needs a few transitions to adjust its amplifier to the correct level.

The reformatted and better commented MANCHESTER code is [available here](#).

Reliable transmission

The Manchester library is really good and over short distances the RF link is good enough that the transmission is very reliably. However, when I have tested moving the TX into another room at the other end of my house I have found the link gets much more lossy and sometimes messages come through corrupted.

~~I don't think there is much more that could be done at the MANCHESTER level.~~ As I wrote this I realised that the sync sequence explicitly checks for high/low frequency interference. However, the data receiving code doesn't do this checking so a clean sync followed by interference will lead to corrupted data. I will need to take a look at improving this.

I decided to add an extra layer of functionality on top of the MANCHESTER library. This extra layer would do the following.

- Send each message 3 times with a random delay between each retransmission.
- Add a message level pre-amble, node ID, message ID and checksum to each transmission.

Blog Archive

▼ 2012 (54)

► October (1)

► June (2)

► May (7)

► April (4)

► March (6)

► February (13)

▼ January (21)

[Playing with LEDs](#)

[Wireless sensor node - turning off subsystems \(5\)](#)

[Wireless sensor node - code management](#)

[Wireless sensor node - sleeping \(4\)](#)

[Wireless sensor node - code fixes \(3 - redux\)](#)

[Wireless sensor node - manchester lib \(3\)](#)

[A couple of interesting links](#)

[Wireless sensor node - rf link working \(2\)](#)

[Wireless sensor node - all on one breadboard \(1\)](#)

[DIY Drones](#)

[Parrot AR Drone - Ideas](#)

[Home Wireless Sensor Network - Battery Life](#)

[Home Wireless Sensor Network - Making Plans](#)

[Time Lapse - Pigs 2](#)

[Time Lapse - Pigs](#)

[Time Lapse Camera](#)

The detailed message format is best described by the code I wrote to send it.

```
// Send a message with the following format

// 6 bits pre-amble

// 5 bit node ID

// 5 bit reading number

// 16 bit data

// 16 bit data (repeated)

//

// This is a total of 3x unsigned ints

unsigned int preamble = (0b010101 << 10);

unsigned int nodeID = ((NODE_ID & 0b11111) << 5);

unsigned int firstByte = preamble | nodeID | (msgNum & 0b11111);

MANCHESTER.Transmit(firstByte);

MANCHESTER.Transmit(data);

MANCHESTER.Transmit(data);
```

Since I only have a single unsigned int of data, sending the data twice is essentially an XOR checksum. I previously XOR'd the data byte with the firstByte but this caused problems because for any data values with the top 6 bits set to 0, the checksum ended up starting with the 6 bit pre-embedable which confused the RX code.

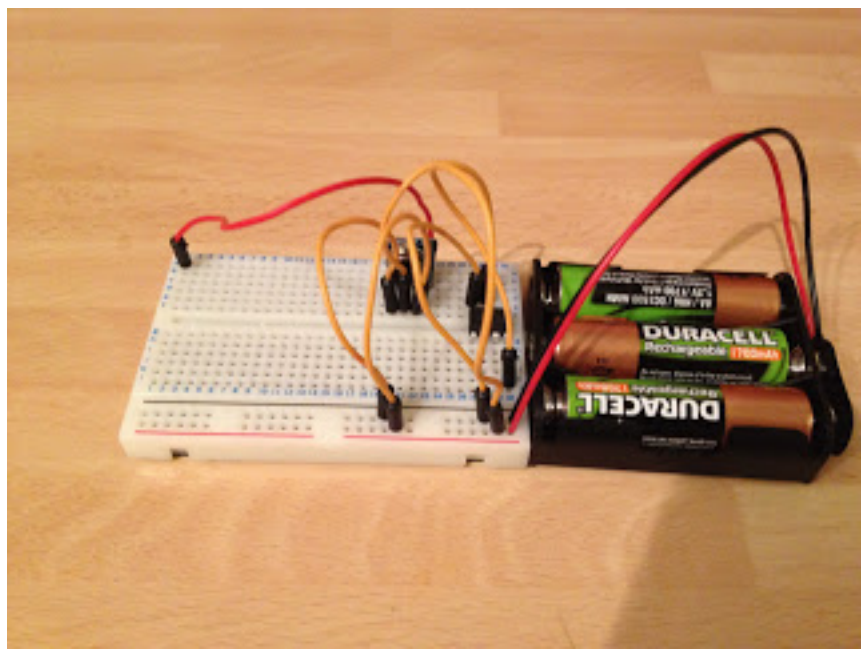
The RX code stores an array of node ID to message Num's. This allows it to ignore retransmissions. The RX code also checks for the pre-embedable, checks the checksum, ~~checks the nodeID and readingNum are within the expected range~~. As I write this I have realised that these last two checks are silly because these numbers are constrained to 5 bits each so actually HAVE to be in the correct expected range.

The full code for my TX and RX code can be seen here:

- [TX](#)
- [RX](#)

Pictures

Every blog post should have pictures. I will conclude this post by pointing out a mistake I made when wiring my circuit up again last night.



- [History - Getting Started With ATTiny85](#)
- [History - Windows Temp Download UI](#)
- [History - Bluetooth temp sensor - Redux](#)
- [History - Bluetooth temp sensor](#)
- [History - Intro](#)

My Other Sites

- [Mchr3k Coding Blog](#)
- [Mchr3k Projects](#)
- www.guineapigdreams.co.uk

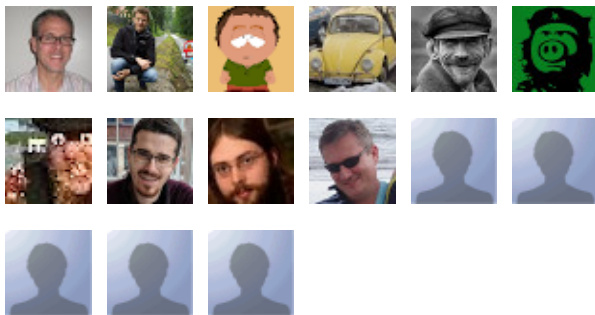
About Me



[View my complete profile](#)

Followers






Faste læsere (15)




Fast læser

This is the small breadboard I use for my TX circuit. I had real issues getting my TX to work last night. I eventually realised that the broken red line along the bottom means that the left half and right half of the power line are disconnected. I initially had my battery plugged in on the left.where it was powering nothing. Moving it to the right got everything working!

Posted by [mchr](#) at 18:08



10 comments:



**Hawk** 22 January 2012 at 19:49

I've also independently taken the time to indent and take a look at the Manchester code. I want it to do something slightly different to you, so I suspect we'll diverge.

Anyway, did you spot there is a bug in the receive pre-amble checking? It checks counthigh again after receiving the low signal. Remember when fixing that the low could last up to a double count + error if it's the end of the pre-amble. I'll blog on it soon (though probably not this evening).



[Reply](#)

▼ [Replies](#)

**mchr**  22 January 2012 at 20:01

Good point - I hadn't spotted that!


[Reply](#)

**mchr**  22 January 2012 at 21:02

All the issues in this post have now been fixed: <http://mchr3k-arduino.blogspot.com/2012/01/wireless-sensor-node-part-3-redux.html>

[Reply](#)

▼ [Replies](#)

**METİN KARGACI** 4 November 2016 at 06:34

Hi,

Links are not working. Where is the Last version of your library? Plase sare your New link.


I have a question. I want to send 2 start bit 16 data bit and 1stop bit. And the I want to receive A data which is included 2 start bit 8data bit and 1stop bit from Slave device. Slave device talks 1200 baudrate.

I tried this with arduino and dali balast(Slave device). But I cant receive any data. But your library working two arduinos. They talk eachother. How can I fix this problem of communication between arduino and Slave device.

Please share your idea.


Thanks.

[Reply](#)

**Unknown** 21 October 2012 at 22:19

This comment has been removed by the author.

[Reply](#)


**maprri** 23 June 2014 at 11:51

Nice library!

A quick question: In Manchester.h, am I correct in assuming 16Mhz for the calculation of HALF_BIT_INTERVAL? I'm running an attiny85 @ 8Mhz, do I need to modify this value?

Thanks.

[Reply](#)

**bule** 3 February 2016 at 07:16

Hi

I am trying to build very small node and lot of them (more then 32) for my smart house. I realy like your library, but I am not sure how will work with one receiver and many nodes. I cant (or dont know how)to set to all nodes send at different time so the receiver wont colapse :-)

Please help me if you can

[Reply](#)

▼ [Replies](#)



mchr [3 February 2016 at 21:14](#)

Unfortunately my library doesn't do anything to help you with this. You will have to try something like sending each message multiple times or attempt to synchronize the nodes so that each one transmits at a different time. Good luck!



Unknown [3 September 2016 at 12:45](#)

use individual node addressing and master-slave polling to talk to each node individually. It's probably the easiest to implement

[Reply](#)



METİN KARGACI [7 November 2016 at 17:15](#)

Hi,

Links are not working. Where is the Last version of your library? Plase share your New link.

I have a question. I want to send 2 start bit 16 data bit and 1 stop bit. And the I want to receive A data which is included 2 start bit 8data bit and 1 stop bit from Slave device. Slave device talks 1200 baudrate.

I tried this with arduino and dali balast(Slave device). But I cant receive any data. But your library working two arduinos. They talk eachother.
How can I fix this problem of communication between arduino and Slave device.

Please share your idea.

Thanks.

[Reply](#)

Enter your comment...



Comment as:

Google Accoun

Publish

Preview

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)