**IR Remote Documentation**

Signal Codes:

Preamble: 0x**fe01**

Power: 0x**fc03**

Speed: 0x**f807**

Rotate: 0x**ec13**

Timer: 0x**f40b**

Mode: 0x**f00f**

Signal Specifications:

See Excel spreadsheet

**Pseudocode**

Configure each I/O space register

Store each signal using 2 registers

//sleep mode is entered, then when interrupt happens, interrupt vector is taken, then program is returned from interrupt with (RETI), program returns to main program (this loop) to reenter sleep mode

Infinite Loop

Enter sleep mode

Jump to before loop //this will protect against accidentally skipping sleep on interrupt return

//pin change interrupt vector is program address: $002

If pin change interrupt happens

Jump to scan: sub-routine from interrupt vector

//pin change sub routine

scan:

store status of SREG

set pinscans to 0xff

scan1:

if PINB = 0x01

send timer signal as argument to output sub-routine

elif PINB = 0x02

send mode signal as argument to output sub-routine

elif PINB = 0x08

send speed signal as argument to output sub-routine

elif PINB = 0x10

send rotation signal as argument to output sub-routine

else

send power signal as argument to output sub-routine

decrement pinscans

if pinscans does not equal zero

jump to scan1

else

jump to exit:

//send output signal, signal code taken as argument

Output:

Set finestart

Set mediumstart

Call modulator

send Preamble through encoder

Loop 3 times

Send signal 1 with encoder:

Send signal 2 with encoder:

Send stop bit with modulator:

Return from sub-routine (to scan sub-routine)

//encodes one bit at a timer, signal/preamble as argument

//encoderbyte for high byte, encoderbyteL for low byte

Mediumencoder:

Restore fencodeloop to 8

Set encoderbyte to signalL

encoder:

shift left bit to carry

set medium to mediumspacemark

set fine to finemark

Call modulator

If carry is set

Set finedelay to finehigh

Set mediumdelay to mediumhigh

Else

Set finedelay to finelow

Set mediumdelay to mediumlow

Jump to delay:

Delay:

Copy fineloop value into temp

Mediumloop:

Restore fineloop value

Fineloop:

Decrement fineloop

If fineloop not equal to zero

Jump to fineloop

Decrement mediumdelay

If mediumdelay not equal to zero:

Jump to mediumloop

Decrement fencodeloop

If fencodeloop does not equal zero

Jump to encoder

Decrement medencodeloop

If medencodeloop does not equal zero

Jump to mediumencoder

//takes in fine and medium parameters, modulates high bits/pulses

modulator:

Copy fine value into temp

Magain:

Restore fine value from temp

Fagain:

Toggle bit

Carrier delay

decrement fine

if fine does not equal zero

jump to fagain:

decrement medium

if medium does not equal zero

jump to magain

return from sub-routine

Tasks

1. Maybe move pin: clz instruction to common space in code?
2. Finish signal branches in pin:
3. Maybe make a timing function