Communicate with LCD via Serial

Byte 1: Descriptor Byte

Bytes 2-x: Data bytes

|  |  |
| --- | --- |
| Descriptor Byte | Details |
| 0x00 | RPM…followed by 2 data bytes (LSB first) |
| 0x01 | Coolant Temp…1 byte |
| 0x02 | Oil Pressure 1 byte |
| 0x03 | Fuel pressure 1 byte |
| 0x04 | Lambda 1 byte (send as lambda\*10^1) |
| 0x05 | Wheelspeeds…4 bytes |
| 0x06 | Accel from ecu 1 byte (for testing) |
| 0x07 | Status…enum or char[]??? |
| 0x08 | go to page (1 byte) |
| 0x09 | Pot 1 value |
| 0x0A | Pot 2 value |
| 0x0B | Pot 3 value |
| 0x0C | Test digits…1 byte |

Create/Find a simple serial library

Create/Find a library to interact with lcd

Shift Lights

Library already found…can be tested on Arduino with atmel studio and a strand of lights

Functionality:

All green on neutral and idle

Flashing red on overheat

Special error condition?

Normal operation

Gear indicator

Find a max7219 library

Do some math

Buttons

ISR’s

Pots

ADC

always updating lights

always updating gear

constantly read adc? Interrupt?

CAN input to AVR

#defines for id, offset, length

Read everything and send to lcd on every loop

Pack all can data into one message

Create an m1 build object for the display area…have its own script for can transmit on 10hz or whatever

Rpm, temp, warnings for the display

shift light status

gear position

at90can will wait for this can id and then update all its shit, check adcs and wait again for it

spit out messages back to the ecu?

Special packet defined in m1 build

Special packet defined in atmel studio

Send stuff via custom serial protocol to display

TODO for Steering Wheel v1.1

Verify pinout in schematic of 7 segment

Measure a 5mm led again and make sure 3d print fits it

Right angle or straight out connectors?

Get my library for microclasp connector

Change buttons/pots connector to microclasp

TODO for display v1.1

Verify fpc flip/flop for the millionth time

~~Make sure new fpc connector footprint is the same~~

~~Move it way up for cable length~~

~~Double check programmer pinout~~

~~Order a new ffc cable~~

~~lcd ffc cable pinout was backwards - fixed~~

~~move lower connectors out of the way so the cable actually fits~~

get schwifty

~~make programmer fit usb bub - fixed~~

~~fix mounting hole layout~~

add more silkscreen

~~rearrange connectors a little bit~~

~~more skookum fpc cable - same as the one on the displays~~

/\*

\* CAN\_Demo.cpp

\*

\* Created: 11/6/2016 9:20:17 PM

\* Author : Mizzou Racing

\*/

#include <avr/io.h>

#include "config.h"

#include "can\_lib.h"

#include "can\_lib.c"

#include "can\_drv.h"

#include "can\_drv.c"

#include <util/delay.h>

#include "mlx90614.h"

#include "i2cmaster.h"

#include "mlx90614.c"

#include "twimaster.c"

int main(void)

{

i2c\_init();

can\_init(0);

st\_cmd\_t message;

message.cmd=CMD\_TX;

message.id.*std*=0x80; //my id

U8 data[6];

message.pt\_data = &data[0];

message.ctrl.ide = 0; //- CAN 2.0A

message.dlc = 6; //data length

int outer,inner,middle;

U8 status;

while(1)

{

// status = can\_cmd(&message);

outer = (int)mlx\_getObjTempF(0x18);

middle = (int)mlx\_getObjTempF(0x19);

inner = (int)mlx\_getObjTempF(0x20);

data[0] = (outer<<8)>>8;

data[1] = outer>>8;

data[2] = (middle<<8)>>8;

data[3] = middle>>8;

data[4] = (inner<<8)>>8;

data[5] = inner>>8;

while(can\_cmd(&message) != CAN\_CMD\_ACCEPTED);

while(can\_get\_status(&message) == CAN\_STATUS\_NOT\_COMPLETED);

//\_delay\_ms(10);

}

}