Weekly Status Report

Week 1

EECE 443

MCU TNC Design

David Cain

C00043561

Submittal Date: September 21st, 2020

Accomplished tasks for this week:

- Create reference experiment document
- Create code to read bits from incoming bitstream of data
- Create code to receive packets of AX.25 and store them into a buffer
- Create code to generate and output an AFSK signal by incrementing through a Boolean array
- Ordered prototyping boards for creating our circuits to connect to the TNC

Planned tasks for next week:

- Fix the AFSK generation code due to phase errors in wave shape
- Run thorough testing on code subsystems and record results with the newly created experimenting sheet
- Flesh out more of the AX.25 protocol interpretation
- Pass Kobe some of the perfboards so he can build the input circuit and PTT circuit

Examples of Completed Task

Example of some code to read in bitstreams:

Code Playground - DAC_SINEWAVE/Core/Src/FreqIO.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

DAC_SINEWAVE

FreqIO.h | AX.25.h | main.c

```c
void initProgram();
void toggleMode();

/*
 *  Function to be ran at Tim3 interrupt
 */
void Tim3IT();
/*
 *  Function to be ran at external trigger.
 */
void FreqCounterPinEXTI();

//GENERATING FREQ
//*****************************************************************************
#define TIM2_AUTORELOAD_TX  100         //Timer2 period, used to control DAC and gene
#define TIM3_AUTORELOAD_TX  831         //Timer3 period, used to controller baudrate

#define PI                  3.1415926
#define OUT_AMPL            4096        //Amplitude of output wave. 4096 -> 1.65V Pe
#define LOWF_SAMP           83          //This is the sample count for the low freque
#define HIGHF_SAMP          45          //This is the sample count for the high frequ

extern bool bitStream[10];

extern uint32_t lowFrequency[2*LOWF_SAMP];      //LUT used to store 2 wavelengths of
extern uint32_t highFrequency[2*HIGHF_SAMP];    //LUT used to store 2 wavelengths of

extern bool     midbit;
extern bool     changeMode;

void edit_sineval(uint32_t *sinArray, int arraySize, int waves, float shiftPercent);
void bitToAudio(bool *bitStream, int arraySize,bool direction);
void generateBitstream();
void initOUTData();

//READING FREQ
//*****************************************************************************
#define SAMP_PER_BAUD       2           //Increase sample rate to detect higher frequ
#define TIM2_AUTORELOAD_RX  10000       //Timer2 period, used for determining frequen
#define TIM3_AUTORELOAD_RX  833/SAMP_PER_BAUD//Timer3 period, used to trigger period8

#define PCONVERT        10000000        //f = 1/T, used for converting period to freq
#define HIGHFREQ        2200            //Higher freq to detect w/ afsk
#define LOWFREQ         1200            //Lower freq to detect w/ afsk
#define FREQDEV         400             //Max potential deviation in target frequency

#define BUFFER_SCALE        32          //Scalar for buffer base
#define BIT_BUFF_BASE       16          //Base amount of bits to store
#define RX_BUFFERSIZE       SAMP_PER_BAUD * BIT_BUFF_BASE * BUFFER_SCALE

extern uint32_t periodBuffer[RX_BUFFERSIZE];    //Stores the period values from inter
extern uint32_t bitBuffer[RX_BUFFERSIZE];       //Stores bitstream values
extern uint8_t  sampusecount;                   //Used to determine old data
extern uint16_t periodSaveCount;                //Used to keep track of index to save
extern uint16_t trackBit;                       //Used to keep track of bits being lo
extern uint16_t bitSaveCount;                   //Used to keep track of bit saved to

/*
 *  Function to convert freq to bitstream:
 *      returns 1 if detect higher freq within threshold
 *      returns 0 if detect lower freq within threshold
 *      return if frequency is outside afsk set range
 */
int pertobit(uint32_t inputPeriod);
/*
 *  Function to take period buffer values and loads the next bit into bit buffer.
 *  Also returns the determined bitvalue
 *  0   = 1200Hz
 *  1   = 2200Hz
 *  -1  = Invalid frequency
 */
int loadBit();
/*
```

FreqIO.c | AX.25.c

```c
    return isriag;
}
int streamGet() {
    int byteArray[8];
    int max_octets = AX25_PACKET_MAX/8;
    int octet_count,good_octet;
    bool gotflag;

    //Just do this unless we need to toggle
    while(!changeMode){
        gotflag = false;
        buffer_rdy = false;

        //Slide bits
        for(int i = 0; i < 7; i++){
            byteArray[i] = byteArray[i+1];
        }
        byteArray[7] = loadBit();

        //Detect AX25 flag bytes
        for(int i = 0;i < 8; i++){
            //If the byte isn't lined up, break loop
            if(byteArray[i]!=AX25TBYTE[i]) {
                gotflag = false;
                break;
            }
            //If the loop makes it to the lowest bit, the flag should be lined up
            else if(i==7){
                gotflag = true;
            }
        }

        //Got flag
        if(gotflag){
            AX25_flag = true;
            sprintf(uartData, "Start AX.25 Flag Detected\r\n");
            HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);
            octet_count  = 0;

            //Until AX.25 buffer overflows, continue reading octets
            good_octet = 0;
            while( (good_octet==0) && (octet_count < max_octets) ){
                good_octet = loadOctet(&AX25_temp_buffer[octet_count*8]);
                //sprintf(uartData, "Loaded octet %d\r\n",octet_count);
                //sprintf(uartData, "good_octet: %d\r\n",good_octet);
                //HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);

                octet_count+=1;
            }

            //If ax.25 buffer overflows or an octet was bad, this was a bad packet
            if((octet_count >= max_octets) || (good_octet!=1)){
                sprintf(uartData, "Bad packet! Not enough octets\r\n\n",octet_count);
                HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);
            }
            //
            else if(octet_count == 1){
                sprintf(uartData, "Stop AX.25 Flag Detected\r\n");
                HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);
                sprintf(uartData, "Bad packet! Not enough octetes.\r\n\n",octet_count);
                HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);
            }
            //If ax.25 buffer does not overflow, this was a good packet
            else {
                sprintf(uartData, "Stop AX.25 Flag Detected\r\n\n");
                HAL_UART_Transmit(&huart2, uartData, strlen(uartData), 10);

                buffer_rdy = true;
                return 1;
            }
        }
        //Didn't get flag
        else {
```

Problems | Tasks | Console | Properties

No consoles to display at this time.

Writable | Smart Insert | 329 : 69 : 8439

Serial Output from controller reading an incoming bitstream:

```
Start AX.25 Flag Detected
Stop AX.25 Flag Detected
Bad packet! Not enough octetes.

Start AX.25 Flag Detected
Stop AX.25 Flag Detected
Bad packet! Not enough octetes.

Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 3 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 4 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 5 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 6 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 7 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 8 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected

Running streamGet() 9 time
Start AX.25 Flag Detected
Printing octet =  0  1  0  1  0  1  0  1
Printing octet =  1  0  0  0  0  0  0  1
Printing octet =  0  0  0  0  0  0  0  0
Stop AX.25 Flag Detected


Port closed
```

Experiment Sheet:

TNC TESTING FORM(REV1)	
Leaf on the Tree:	
Device Under Test (Testing Tree Number):	
Date:	
Person(s) Conducting Experiment:	
Signature:	
Experiment Purpose:	
Experiment Procedure:	
ent Settings/Software Settings (w Revision):	
Testing Diagram/Picture:	
Data Points:	
Pass/Fail:	
Interpreted Notes:	
Recommendations for Modification:	

Prototype Board Order Form:

Your order's in!

You should get it by **Oct 1.**

Order total: $7.49

Shipping to:

Order number: **24-05773-92472**

10X Double Side 5x7cm PCB Strip board Printed Circuit Pro...
You should get this by Oct 1.

See order details

Time Sheet

Item	Date/Time	Description	Hours
1	9/14/2020 5:30pm-7pm	Begin creating experiment sheet by overlooking sheet provided by Dr. Darby.	1.5
2	9/14/2020 3:30pm-6pm	Use pre-written bitLoading operation to create function for converting an AFSK digital bitstream based on AX.25 flag detection.	2.5
3	9/16/2020 5:30-7pm	Using the previous digital bitstream conversion software, implement functional logic to create an AX.25 packet.	1.5
4	2/16/2020 12pm-3pm	Create software to output an AFSK bitstream. This was done by storing the bitstream as an array of Booleans and instruct the DAC to output a sine.	3
5	2/17/2020 12pm-1pm	Spend a bit of time looking for good prototyping boards to assemble the circuits on and connect to our microcontroller.	1
		Total:	8.5