Assignment 1 Presentation

EECE 443

Team 2: MCU TNC Design
Kaleb Leon – C00094357
Kobe Keopraseuth – C00092349
David Cain – C00043561

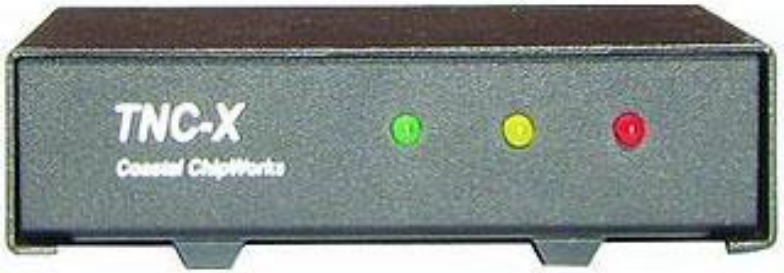
February 18, 2020

MCU TNC Assignment 1

David Cain, Kobe Keopraseuth, and Kaleb Leon

What is a TNC? (Terminal Node Controller)

A device used by amateur radio operators to participate in AX.25 packet radio networks, with the addition of a modem to convert baseband digital signals to audio tones.



How packets are formatted

HDLC (High level Data Link Control)

Flags (start and end) – 01111110

Address – AX.25

Control – What is the data? Information, number, unnumbered, or supervisory

Information – Basically your data up to 256 octets(512 bytes)

FCS(aka Parity Bits) – used for error checking, a set of bits defined at the transmission side from the other frames using algorithm. If not the same on the received side packet is discarded.

Flag	Address	Control	Information	FCS	Flag
8 bits	8 or more bits	8 or 16 bits	Variable length, 8×*n* bits	16 or 32 bits	8 bits

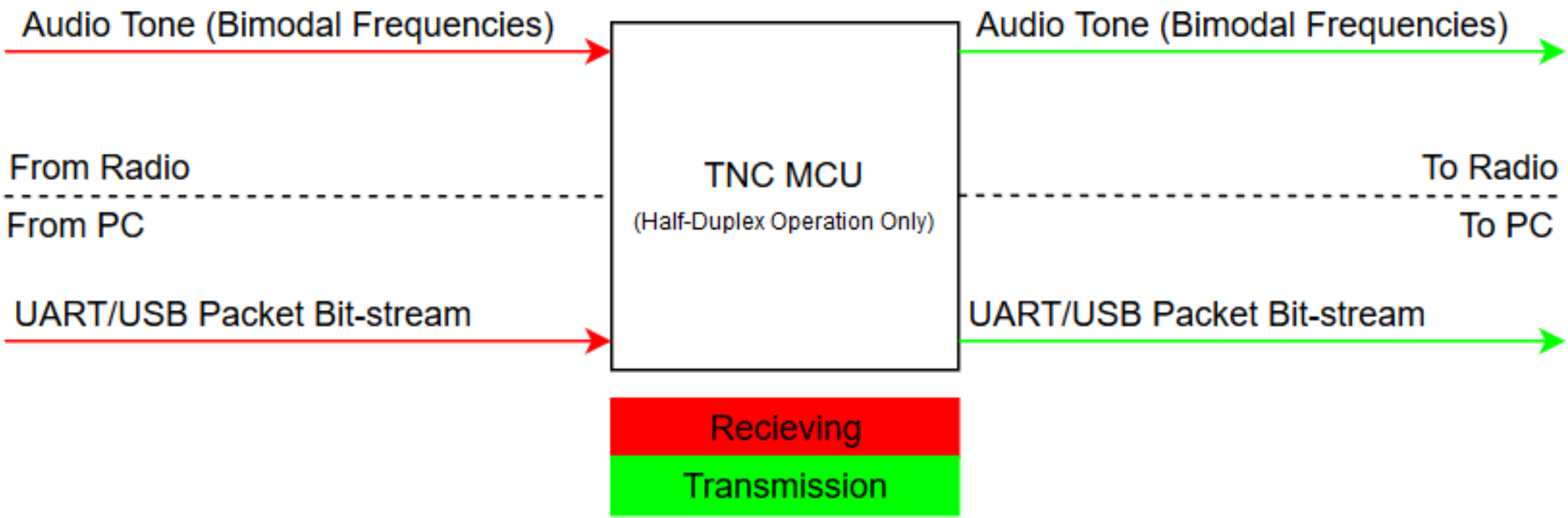
AX.25 Protocol

Specifies that the address field of the HDLC frame be split into multiple addresses:

- Source Address
- Zero or more repeater addresses
- Destination Address

Addresses correspond to stations call signs

Project Level 0 Diagram



Scope of Work

System Basic Functionality

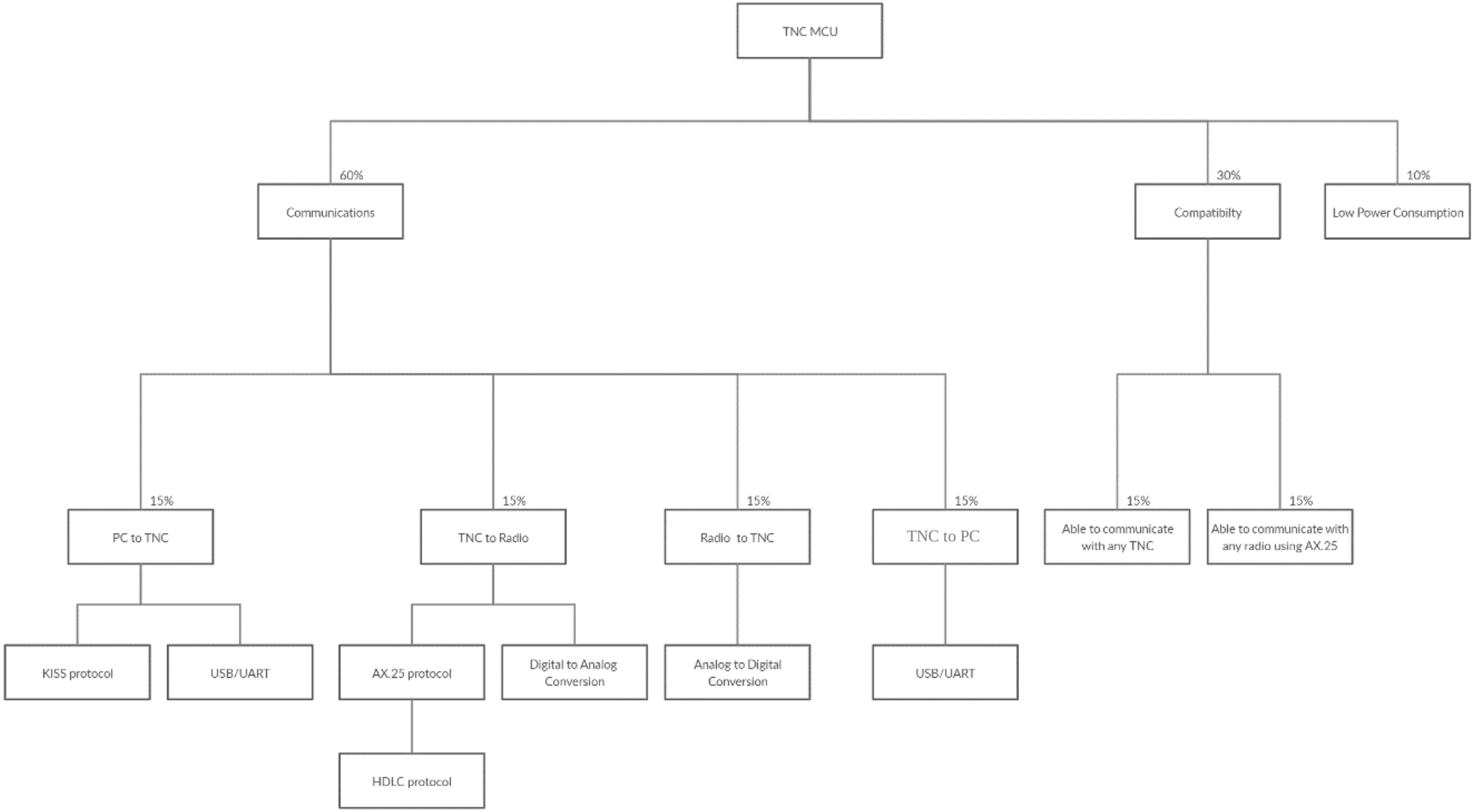
Receiving

1. Receive Audio Tone Signal
2. Analog to digital convert into binary
3. Gather payload
4. Check for Errors
5. Send payload via KISS to PC

Transmitting

1. Receive KISS formatted data from PC
2. Follow AX.25 protocol to form data packet
3. Translate into analog audio tone signal
4. Send Audio tone to radio

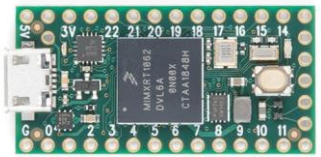
Objective Tree

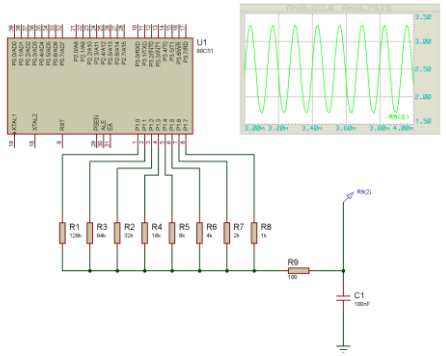


Specs

Category	Requirements
Power	3.3 VDC to 5.5 VDC 10 mA
Environmental	-30 to 70 deg C <2x2 Inches
Audio Input	50 mV into 1K ohms BER 10-3 @ 6db snr
Audio Output	400 mV into 1K ohms
Protocol	KISS Mode AX.25 HDLC
Digital Input	3 Volt UART
Digital Output	3 Volt UART
LED Indicators	PTT indicator RX good Packet Energy in Audio Passband
PTT (Push to Talk)	Active High & Low Supply and Sink 20 mA Accept 3 to 15 Volts

Alternatives and Tradeoffs

MICROCONTROLLER	STM32L4433	Teensey 4.0	Arduino Mega
			
Description	This microcontroller contains 16 external ADC channels, 1 12-bit ADC, 2 12-bit DAC output channels, an on board RTC, 2 CAN buses, 2 ultra-low-power comparators, CRC calculation unit, and a Schmitt trigger I/O.	This microcontroller contains 40 digital pins (all interrupt capable), 14 analog pins, 2 ADCs on chip, a RTC for date/time, an ARM Cortex-M7 at 600 MHz, 1024K RAM (512K is tightly coupled), and a 2048K Flash (64K reserved for recovery & EEPROM emulation).	This microcontroller contains 16 Analog read pins, 53 Digital pins, and 6 interrupt pins.
Cost	**14.90**	**19.99**	**18.99**
Pros	•Contains CRC calculation unit •Low Cost •Many GPIOs	•Fast clock speed •Has RTC	•Easy to use •Many GPIOs
Cons	•Embedded C programming	•Highest Cost •No CRC calculation unit	•Does not contain RTC •Does not contain DACs or ADCs

Circuit Design	Built into Controller	Resistor Switching Network	DAC IC
			
Description	If the design were to include any of the STM32 line, the MCUs have builts in DACs.	Would only consist of using ~4-6 GPIO, connected to different resistor values to represent variable step voltage output. This output would be passed through an LPF to generate a smooth sinusoid.	This would be using a dedicated High-Speed DAC ICs (such as DAC38RF82) that only requires digital input translated to an analog wave for us.
Cost	Similarly to the dedicated IC, the benefit is there will only be a need to generate digital values.	to this option is the simplicity and lack of components needed to generate waveform at low power cost.	is the ease of use, only needing to generate digital values that will quickly be converted to sinusoidal waveform.
Pros	Often built in DACs are slow and this may not work within the strict timing constraints of AX.25	would be the requirement to create code to drive a resistor network meaning more time would be spent on the DAC	With the dedicated silicon, this will raise the price and power consumption of the board.
Cons	If the design were to include any of the STM32 line, the MCUs have builts in DACs.	Would only consist of using ~4-6 GPIO, connected to different resistor values to represent variable step voltage output. This output would be passed through an LPF to generate a smooth sinusoid.	This would be using a dedicated High-Speed DAC ICs (such as DAC38RF82) that only requires digital input translated to an analog wave for us.

Feasibility

Technological Analysis

The hardware for the project has only a few requirements:

- Interface with a terminal program via UART/USB
- Packetize data into AX.25 Protocol requirements
- Detect the frequency of an incoming audio signal
- Output an audio signal representing the packet bitstream
- Or output the packet bitstream via UART/USB

With the goal of the project aiming to replace hardware that was originally developed in the early 1980s, this means the technical requirements are minimal.

Tasks such as interfacing with UART or detecting the frequency of an audio signal not only have many options but also are very well documented.

Time Analysis

- The physical designing and testing will begin after a month of research
- A goal of our project is to ensure documentation is enough for future groups to have a strong understanding of our system
- To ensure this outcome, every three weeks we will dedicate time to ensuring the documentation current at that time is thorough and informative.

Gant Chart for Semester 1