KG7YU remote keyer system.

The remote keyer system is designed to allow the use of remote paddles, and keys (straight keys, bugs, side-swipers) to be used to control an amateur radio transmitter. The system uses the internet to send messages from the remote location to the local location where the transmitter is located. Two controllers are used, a Remote controller connect to the key and the Local controller connected to your transmitter.

At the heart of both of these controllers are Adafruit Feather development boards. These development board are programmed using the Adruino development environment. This environment has broad community support allowing rapid development.

Remote controller

The Remote controller uses an Adafruit Feather HUZZAH development board utilizing a ESP8266 WiFi module. This ESP8266 WiFi module incorporates a microcontroller clocked at 80 MHz. The Remote system reads and processes the contact closures from the attach paddle and key then sends this data using the WiFi link to the connect Local system. Below is a list of the Remote controller functions:

* WiFi network connection
* Opens UDP and TCP ports with local controller
* Built in iambic keyer
* Sidetone generation
* Ability to set key up and key down signals as well as higher level dit and dash signals
* Mixes audio from computer with sidetone
* Blanks computer audio during key operation
* Connection status LED
* Connection button to initiate link
* USB powered
* USB host interface commands to configure and save settings

Local controller

The Local controller uses as Adafruit M0 Feather development board and a Adafruit Ethernet featherwing. The featherwing adds a hardwire ethernet interface. The Local controller is directly connector to the internet access point near the transmitter. The Local controller accepts incoming connections from the Remote controller and processes key signals. The transmitter is keyed using an open collector output that connects to the transmitters key input. Below is a list of the Local controller functions:

* Hardwire ethernet connection
* Accepts UDP and TCP connections from the remote controller
* Processes key up and down as well as paddle messages to key the transmitter
* Link performance testing
* Fail safe max key down limits
* Auxiliary control outputs
* USB powered
* USB host interface commands to configure and save settings

The access point at the local location needs to be configured to forward the UDP and TCP ports to the IP address if the Local controller. The Local controller should also be assigned a fixed IP address. Commands in the Local controller allow you to define and save the IP address. This setup only needs to be done once.

The Remote controller needs the SSID and password for the wireless access point you are using. You can even tether to a hotspot from a smartphone. After you are connected to the WiFi access point the local IP address is needed to make the connection. Most home internet links do not have a static IP but its pretty easy to get your IP and it will not change very often. Control of the station is done using PC remote control application and one easy way to get your local IP address is to open a browser and enter My IP in the search box.

Operating my Amateur radio station remotely has been a goal for some time. I have slowing been moving in this direction by enabling control of my station through my control PC. Using TeamViewer I can log in to the PC in my shack and have full control of everything expect the ability to use a remote key. I looked at number of remote key option but never found anything that provided all the features I wanted. My goal is a remote keying option that supports both paddles as well as straight key mode that will support SKCC operation. Think of this as a virtual wire from my remote location to my shack where key contact closures are sent with minimal processing.

The solution I have developed uses the internet to send data packets from my remote location to my shack to key the transmitter. The heart of the remote key system are Adafruit Feather development modules programmed using the Arduino development environment. I use a WiFi module for the remote system and a hardwired ethernet module for the local system. The key down and up data is sent using UDP data packets, these packets have lower latency but they have a higher likely hood of being dropped or missed. The latency is not as critical as the consistency of the latency, basically the latency jitter will have a negative impact of the reconstructed key signals at the local system. The first step was a simple link test where UDP packets are sent every 100 mS from the remote system to the local system, the local system then reports the received time delta between packets. The results were encouraging and are posted on my github account. I tested at a couple locations and also used by iPhone as a remote access point. This test convinced me that the concept would work.

The Remote Key system uses two modules; a Local controller in my shack connected to the ethernet and my transmitter, and a Remote controller that connects to my key and computer.

The Local controller uses as Adafruit M0 Feather development board and a Adafruit Ethernet featherwing. The featherwing adds a hardwire ethernet interface. The Local controller is directly connected to the internet access point near the transmitter. The Local controller accepts incoming connections from the Remote controller and processes key signals. The transmitter is keyed using an open collector output that connects to the transmitters key input. Below is a list of the Local controller features:

* Hardwire ethernet connection
* Accepts UDP and TCP connections from the remote controller
* Processes key up and down as well as paddle messages to key the transmitter
* Link performance testing
* Fail safe max key down limits
* Auxiliary control outputs
* USB powered
* USB host interface commands to configure and save settings

The Remote controller uses an Adafruit Feather HUZZAH development board utilizing a ESP8266 WiFi module. This ESP8266 WiFi module incorporates a microcontroller clocked at 80 MHz. The Remote system reads and processes the contact closures from the attach paddle and key then sends this data using the WiFi link to the connect Local system. Below is a list of the Remote controller features:

* WiFi network connection
* Opens UDP and TCP ports with local controller
* Built in iambic keyer
* Sidetone generation
* Ability to send key up and key down signals as well as higher level dit and dash signals
* Mixes audio from computer with sidetone
* Blanks computer audio during key operation
* Connection status LED
* Connection button to initiate link
* USB powered
* USB host interface commands to configure and save settings