Setting up the wireless timer and speakers

# Overview

This document describes how to use Raspberry Pi Zero W devices to create a portable WiFi network used to drive one or more remote LED timer displays and/or remote speaker systems.

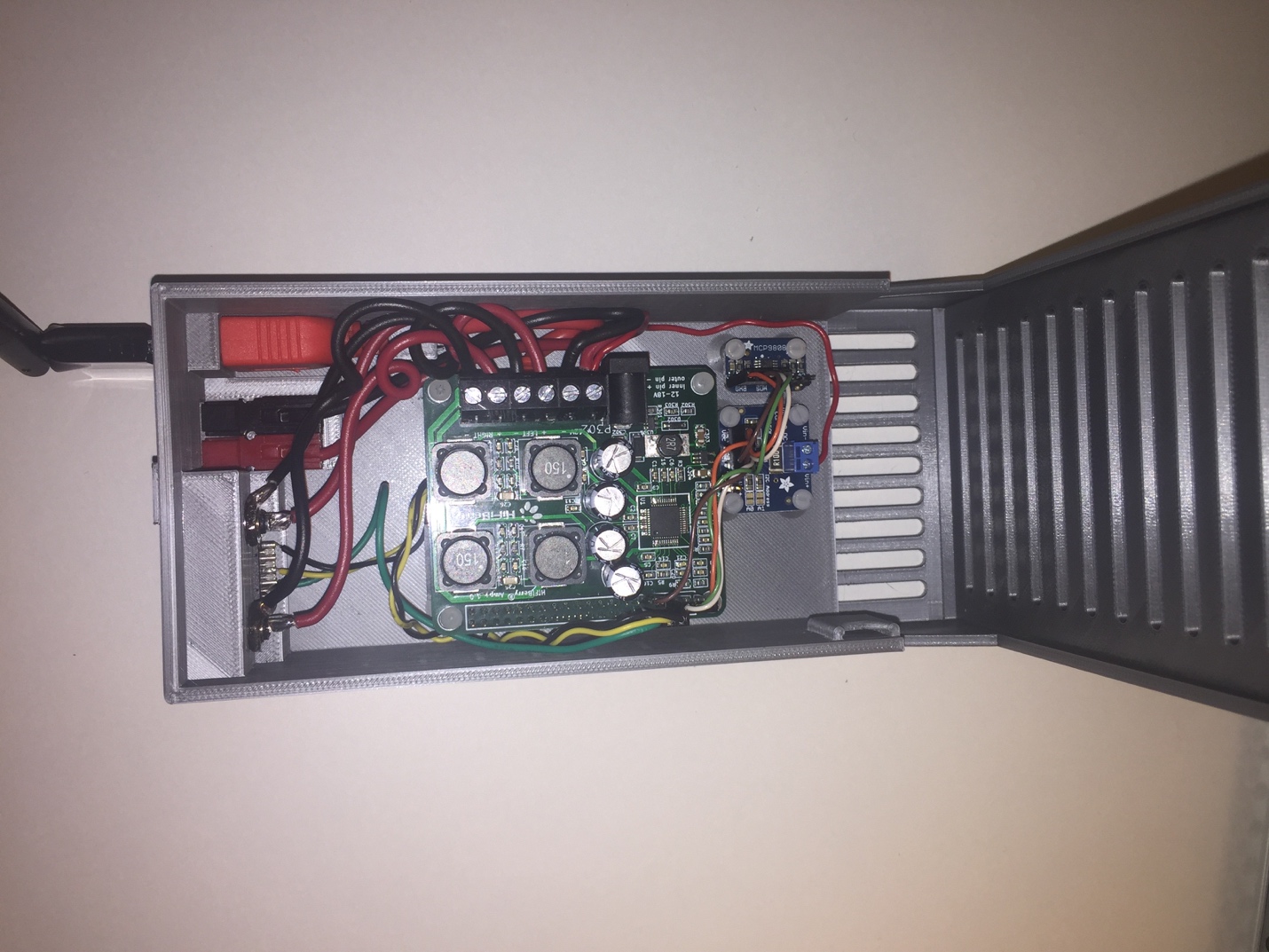
Required hardware / software

# Scoring PC

* GliderScore (<http://gliderscore.com>)
* GliderScoreRemote (software from David Beach)
* TuneBlade software (<http://www.tuneblade.com>)
* VB-Cable software (optional, only if necessary – https://www.vb-audio.com/Cable)
* WiFi adapter (**One Pi will access as a wireless access point**)
* Optional – cell phone either direct cabled or Bluetooth tethered for internet access

# Recommended Pi hardware

* Raspberry Pi Zero W
* Two micro SD cards (one configured as access point, the other as remote node)
* Adafruit P1782 temperature sensor
* Adafruit P904 voltage sensor
* Adafruit P1020 USB WiFi adapter
* Sparkfun USB OTG Cable (Female A to Micro B – 5”)
* HiFiBerry AMP+ or AMP2 or DAC+ Zero (requires external amplifier)
* Anderson PowerPole 30A connectors
* RCA style speaker connectors
* Outdoor public address speaker horns





# Building the Pi Zero Server on Stretch Lite (headless with just a network and some luck!)

* Note: Do NOT plug in the USB wifi adapter yet!
* Generally we follow the guidelines given at <https://learn.adafruit.com/raspberry-pi-zero-creation/overview>
  + Download Raspian Lite (this assumes Stretch release 2017/11/29)
  + Use an image utility to prepare an SD card.
  + Create the following files in /boot
    - **./wpa\_supplicant.conf** (configured for your home wireless network)

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev  
update\_config=1  
   
network={  
 ssid="<your network id>"  
 psk="<your network password>"  
 scan\_ssid=1  
}

* + - **./ssh** (empty file enabling ssh at boot)
  + Edit **/boot/config.txt** – (add to the end of the file)

#enable UART (hopefully we won’t need it)  
enable\_uart=1

* VERIFY your edits – you won’t have network access unless these files have the correct names and content.
  + Changed files in /boot:
    - wpa\_supplicant – contains your home wireless name and password
    - ssh – an empty file
    - config.txt – last line is enable\_uart=1
* Power up your PI and hope for the best. If all goes well you’ll have the pi boot, network enabled with ssh access. The defaults are machine name raspberrypi, user name pi, password raspberry.
  + ssh –l pi raspberrypi (or raspberrypi.local or via ip address located via router)
  + nano .bashrc
    - sudo bash (add this to the end of the file – you will be root every time you log in
  + source .bashrc (you should now be root)
  + **nano /etc/systemd/system.conf (give fsck enough time to complete at boot!)**

DefaultTimeoutStartSec=180s  
DefaultTimeoutStopSec=180s

* + Use raspi-config
    - Change User Password
    - Network Options – change host name
    - Boot Options – enable wait for network
    - Localization options - update all localization options
    - Interfacing options - enable ssh, i2c, serial
    - reboot (be patient this boot expands the root partition to available disk space)
* Now bring all the software and hardware up to latest releases. NOTE: this login may require ssh host reference by IP address. (TODO: why is IP address required for this ssh login?)
  + apt-get update
  + apt-get upgrade -y
  + reboot
  + apt-get dist-upgrade -y
  + rpi-update
  + reboot
* Now configure for dual wifi adapters
  + in /etc/wpa\_supplicant  
    mv wpa\_supplicant.conf to wpa\_supplicant-wlan0.conf  
    \*\*\* NOTE: get the name correct (with a dash) or you will lose network access \*\*\*
  + nano /etc/dhcpcd.conf (append the lines below)

# add a static ip for wlan1  
interface wlan1  
static ip\_address=10.0.0.1/24  
static routers=10.0.0.1  
static domain\_name\_servers=10.0.0.1

* + plug in the USB WiFi adapter now
  + reboot
* Install the access point and dhcp server software
  + apt-get update \*\*\* note: do it again or hostapd will not be found!
  + apt-get install hostapd dnsmasq -y
  + nano /ect/dnsmasq.conf – append the following:

interface=wlan1  
dhcp-range=10.0.0.2,10.0.0.255,255.255.255.0,24h

* + nano /etc/default/hostapd

DAEMON\_CONF="/etc/hostapd/hostapd.conf"

* + nano /etc/hostapd/hostapd.conf (it is a new file)

interface=wlan1  
hw\_mode=g  
channel=10  
auth\_algs=1  
wpa=2  
wpa\_key\_mgmt=WPA-PSK  
wpa\_pairwise=CCMP  
rsn\_pairwise=CCMP  
wpa\_passphrase=ifly4fun  
ssid=ESL-NET  
ieee80211n=1  
wmm\_enabled=1  
ht\_capab=[HT40][SHORT-GI-20][DSSS\_CCK-40]

* + systemctl enable hostapd
  + systemctl enable dnsmasq
  + reboot
* Configure the sound card per instructions at <http://www.hifiberry.com>
  + Remove the line ‘dtparam=audio=on’ from /boot/config.txt if it exists
  + Add the appropriate driver for your sound device in /boot/config.txt  
    dtoverlay=hifiberry-dac (or hifiberry-dacplus or hifiberry-amp)
  + Create /etc/asound.conf with the following content

pcm.!default {  
 type hw card 0  
}  
ctl.!default {  
 type hw card 0  
}

* + reboot
* Install shairport-sync
  + apt-get install shairport-sync
  + reboot
* That’s it for now. Install TuneBlade (and optionally VB-Cable if needed) and try it out!

Installing udpTelemetryDriver and udpTimerDriver

* Remove ‘console=serial0,115200’ from /boot/cmdline.txt
* Download the executable files (udpTelemetrySender & udpTimerDriver) from GitHub and copy them to /usr/bin
  + wget <https://github.com/dsbeach/GliderScoreRemote/raw/master/Bin/Raspbian/udpTimerDriver>
  + wget [https://github.com/dsbeach/GliderScoreRemote/raw/master/Bin/Raspbian/udpTelemetrySender](wget%20https:/github.com/dsbeach/GliderScoreRemote/raw/master/Bin/Raspbian/udpTelemetrySender)
* chmod +x /usr/bin/udp\*
* nano /lib/systemd/system/udpTelemetrySender.service

[Unit]

Description=ESL udpTelemetrySender Service

After=multi-user.target

[Service]

ExecStart=/usr/bin/udpTelemetrySender

WorkingDirectory=/home/pi/

StandardOutput=inherit

StandardError=inherit

Restart=always

User=root

[Install]

WantedBy=default.target

* nano /lib/systemd/system/udpTimerDriver.service

[Unit]

Description=ESL udpTimerDriver Service

After=multi-user.target

[Service]

ExecStart=/usr/bin/udpTimerDriver

WorkingDirectory=/home/pi/

StandardOutput=inherit

StandardError=inherit

Restart=always

User=root

[Install]

WantedBy=default.target

* systemctl enable udpTelemetrySender.service
* systemctl enable udpTimerDriver.service
* reboot

Installing the web server and php

* apt-get install apache2 php7.0
* reboot

# Refreshing the raspian binaries.

* Copy the code below to each pi as file named ‘refresh’.

#!/bin/bash  
systemctl stop udpTelemetrySender  
systemctl stop udpTimerDriver  
**rm** udpTelemetrySender  
**rm** udpTimerDriver  
**rm** /usr/bin/udpTelemetrySender  
**rm** /usr/bin/udpTimerDriver  
wget <https://github.com/dsbeach/GliderScoreRemote/raw/master/Bin/Raspbian/udpTelemetrySender>  
wget <https://github.com/dsbeach/GliderScoreRemote/raw/master/Bin/Raspbian/udpTimerDriver>  
**mv** udpTelemetrySender /usr/bin  
**mv** udpTimerDriver /usr/bin  
**chmod** +x /usr/bin/udpTelemetrySender  
**chmod** +x /usr/bin/udpTimerDriver  
systemctl start udpTelemetrySender  
systemctl start udpTimerDriver

* chmod +x refresh
* ./refresh

# Converting from an access point to a node.

* Create wpa\_supplicant-wlan1.conf in /etc/wpa\_supplicant

country=US

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

network={

ssid="ESL-NET"

psk="ifly4fun"

scan\_ssid=1

}

* Remove the static IP info from /etc/dhcpcd.conf (# makes it a comment)

# add a static ip for wlan1

#interface wlan1

#static ip\_address=10.0.0.1/24

#static routers=10.0.0.1

#static domain\_name\_servers=10.0.0.1

* Disable the access point and dhcp server services

systemctl disable hostapd

systemctl disable dnsmasq

* Reboot – note that you can boot up a node first, and it should automatically attach once the access point comes up.

Random notes for Pi related stuff

* Disk to disk SD cloning on a mac
  + Plug in the source disk first
  + Plug in the destination disk
  + Confirm devices using gui Disk Utility
  + Unmount source and destination volumes (using the gui)
  + Say the magic word:

sudo dd if=/dev/rdisk2 of=/dev/rdisk3 bs=65536 conv=noerror,sync

* Temporarily shutdown a network interface on a pi
  + sudo ip link set wlan0 down (use up to bring it back or reboot)