Ham 97C - ASV Control - Pi Python GPIO

Dr. Marc & Rosemary © 230815

1. Connections to the outside world can be very useful to turn on a light, lock a door, or see if a door is closed.
   The radio VOIP is Asterisk telephone PBX, so messages cannot be displayed. Use Bash script inside Python to speak.

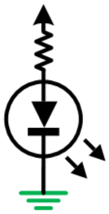
2. Use HamVoip to call Python for controlling Raspberry LED. Any low voltage interface relay substitutes for LED.
   a. Parts needed: LED, 330 Ohm (Ω) resistor, two jumper wires
   b. Connect the anode long leg to the resistor. Other side of resistor to positive supply. Use Pin 18 (6 down on right)
   c. Connect the cathode flat side to ground. (3 down on right.)
   d. The pin ratings are 3.3 VDC with limited current. Be careful with 5V TTL. Input has default pull-up resistor.

3. PuTTY into the node.
   a. Enter the hostname / IP address. Change to Port 222. Click SSH. Give a Session-Name. Click Save. Click Open.
   b. A small black screen shows connected to Pi's Linux CLI. Enter Linux login: root and password: YourName.

4. HamVoip Admin Menu grey screen opens. Menu1 Updates are a good idea for compatibility before additions.
   a. Menu 9- Start Bash Shell to access the Pi Linux CLI. A black screen shows you made it.
   b. Menu 11- Run Asterisk CLI. At black screen, type '!' to get to Linux CLI.

5. Add Python-install-pkg, if not installed with Skywarn. HamVoip is outdated Linux Python versions. Ignore warning.
   a. Copy each line. On Pi, paste at Linux prompt. <enter>. Accept the defaults.

```
wget https://bootstrap.pypa.io/pip/3.5/get-pip.py
python get-pip.py
```



6. To Access Pi pins, add Python GPIO package.

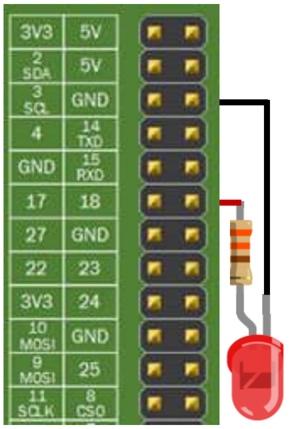
```
python3 -m pip install RPi.GPIO    ; Python-install-pkg puts GPIO into Python
```

   a. Personal scripts store at */etc/asterisk/local*, so they are not overwritten. Change directory.

```
cd /etc/asterisk/local             ;directory for personal scripts
ls -l                              ; ls files with permissions
```

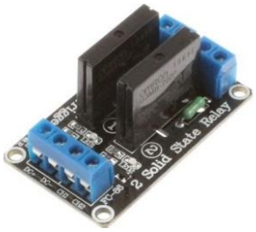
7. Create or edit a Python file to access Raspberry GPIO.

```
nano ModLed.py                     ;  nano editor. Names are case sensitive.
```



8. Copy the code. Paste in the file. This is fixed space Consolas font, so it can be copied.

```
#!/bin/python                      # Python file
# Dr. Marc Durham, NM0D, 230815
# Set output pin, speak its state, then read its state.
import RPi.GPIO as GPIO            # Python library GPIO pin
import time                        # Python library time
import subprocess                  # allows running Bash from Python
GPIO.setmode(GPIO.BCM)             # Broadcom (BCM) pin naming convention
GPIO.setwarnings(False)            # turn off warnings
GPIO.setup(18,GPIO.OUT)            # set pin 18 as output
GPIO.output(18,GPIO.HIGH)          # make output high = 3.3V
subprocess.run(['speaktext.sh', 'on', '58000'], cwd='/usr/local/sbin')
time.sleep(2)                      # call sleep timer for 2 second
GPIO.output(18,GPIO.LOW)           # make output low = 0V
subprocess.run(['speaktext.sh', 'off', '58000'], cwd='/usr/local/sbin')
state = GPIO.input(18)             # get input from the pin
```



9. Set executable properties as 'rwx r-x r-x'. The binary numerically translates to 755. At Linux prompt, type:

```
chmod 755 ModLed.py                ;  give permissions
```

10. Test the file while still in Bash shell, any folder.

```
python /etc/asterisk/local/ModLed.py          ; sudo not needed if root user.
```

11. To make an event in Asterisk cause the file to execute, put it in a */etc/asterisk/rpt.conf* stanza. In this case, button D1.

```
[functions58000]                              ;  controls DTMF ops
D1=cmd,/usr/local/sbin/saytime.pl 74008 58000 ;  change node numbers 4 places
D2=cmd,python /etc/asterisk/local/ModLed.py -rx 58000    ; Pi GPIO.
```

12. Interim actions: <Ctrl> x, Y(es), <enter> leaves nano. 'exit' returns to Admin Menu.

13. Menu 13- Restart Asterisk.

14. Life is good. Enjoy!

Web: evergreencq.org Social: gab.com/groups/62710