CIRCUIT NOTEBOOK

Interesting circuit ideas which we have checked but not built and tested. Contributions will be paid for at standard rates. All submissions should include full name, address & phone number.

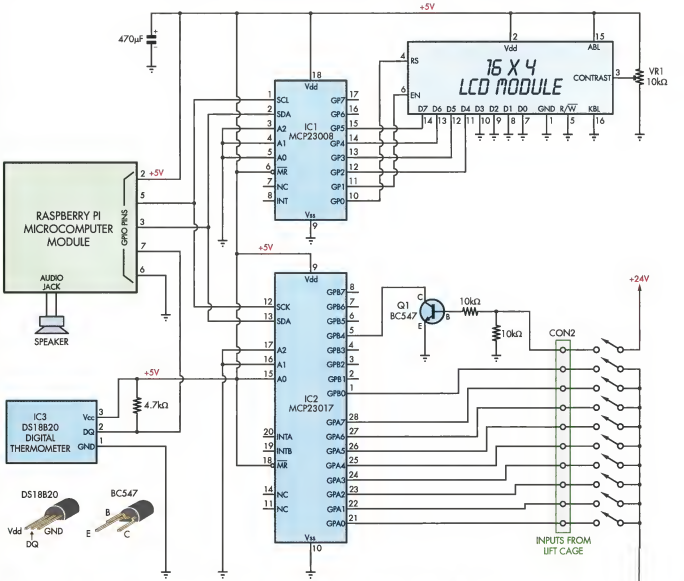
Raspberry Pi Elevator Display & Annunciator

The maintenance crew at my workplace got sick of replacing the incandescent globes which indicate the current floor in our lifts. These are special 24V bulbs which are now becoming quite difficult to obtain and expensive.

Rather than persevere with the old floor level indicators, I decided to completely replace the display with a 16x4 backlit alphanumeric LCD driven by a Raspberry Pi. In addition to displaying the current floor, the LCD also shows the time, date and the current temperature. As well, with the addition of a speaker, it can announce the floor numbers.

The circuit can be used in any application which requires the detection of changing voltages and the software can be easily modified to make different kinds of announcements.

The circuit makes use of two input/output (I/O) expander ICs in order to interface with a standard 16x4 alphanumeric LCD and switches in the lift cage which indicate the current floor. The Raspberry Pi's I/O



SC ©2017 RASPBERRY PI ELEVATOR DISPLAY & ANNUNCIATOR

pins can only tolerate 3.3V, hence the use of the 5V I/O expander IC1 to communicate with the 5V LCD. IC2 could run off either 3.3V or 5V as it's primarily intended to detect contacts closing to ground.

IC1's GP0-GP5 pins are configured as outputs and these connect to the reset (RS), enable (EN) and data (D4-D7) pins of the LCD respectively. The Raspberry Pi controls IC1 over a 2-wire I^2C bus, connected in parallel to the SDA (data) and SCL (clock) pins of both IC1 and IC2 and to GPI/O pins 3 and 5 on the RPi.

IC1 and IC2 have different I^2C addresses, due to the differing configuration of their A0-A2 address inputs, hence the RPi can address them one at a time. Since A0-A2 of IC1 are connected to ground, its address is 20 (hexadecimal) while A0 of IC2 is tied to +5V, giving it an address of 21 (hexadecimal).

The MCP23008 (8-bit) and MCP23017 (16-bit) I/O expanders, like many micros, have internal configurable weak pull-up currents which can be enabled or disabled for each I/O pin. These are utilised to sense when the lift cage switches pull those pins to ground.

Most lifts will close contacts to ground, and nine inputs of IC2 are shown connected directly to these switches. However, some lifts may have switches which connect the pin to a +24V bus instead. Transistor Q1 is configured to invert this and pull pin GPB4 (pin 5) of IC2 low when that input goes high. In some applications, you may need a similarly connected transistor for each I/O pin.

Besides the speaker connected to its audio jack (to make voice announcements) and the 5V regulated power supply (not shown), the only other hardware attached to the Raspberry Pi is a 1-wire digital temperature sensor (IC3) so that the LCD can show the ambient temperature in the lift.

Editor's note: we've explained how this sensor can be attached to a Raspberry Pi in past articles; see the March 2016 issue on pages 34-37; www.siliconchip.com.au/Issue/2016/May/4-Input+Temperature+Sensor+PCB+For+The+Raspberry+Pi

Note that you could use a passive high-sensitivity speaker for this project, such as the types fitted with 3.5mm jacks and intended for use with computers and tablets. If you want to use a regular speaker, you will need a small amplifier and suitable power supply.

The software

The software that runs on the Raspberry Pi is written in Python. There are two different versions of the software. One plays an MP3 file each time the lift moves to another floor. The other uses speech synthesis to provide the announcements.

The former will provide a more natural sound while the latter avoids the requirement to record a series of MP3 files and may prove more flexible if adapted for other applications, where more complex announcements may be required.

The main Python script is launched from the /etc/rc.local script so that it runs at boot time. It constantly monitors the inputs of IC2 and if the states change, it either launches mpg321 to play an MP3 file (first version) or espeak to produce the synthesised voice (second version).

The software requires the free libraries from Adafruit called "Adafruit_I2C.py", "lcd23008.py" and "Adafruit_MCP230xx.py" to drive the LCD and the I/O expander ICs.

These are included in the download package and should be placed in the same directory as the main script ("lift_light.py" or "lift-light-with-espeak.py"), both of which are available from the SILICON CHIP website.

The following steps are required to set up the software:

1. run **sudo apt-get update**
2. run **sudo apt-get upgrade** to bring your Raspberry Pi software up to the latest version
3. run **sudo apt-get install alsa-utils** to install the sound utilities.
4. run **sudo 'echo snd_bcm2835>> /etc/modules'** to load the sound module, then reboot the RPi (**sudo reboot**) to enable sound
5. run **sudo apt-get install mplayer espeak espeak-gui** to install the MP3 player and voice synthesiser software

6a. if using the MP3-based version, record an MP3 for each floor named "01.mp3", "02.mp3", "03.mp3", etc and copy them to the RPi. Edit the script "lift_light.py" and change the paths after each invocation of mpg321 to refer to these files (sample MP3 files are included in the download package)

7b. if using the voice synthesised version, run **espeak-gui** and choose the required language (eg, English). Once finished, you can test it from the command line by running the following commands (a=male voice, b=female voice):

a. **espeak "Hello from Silicon Chip"**

b. **espeak -ven+f3 "Hello from Silicon Chip"**

7. if you are using a version 2 Raspberry Pi, edit the script and change this line:

```
lcd = Adafruit_CharLCDPlate(
                busnum = 0)
```

to use: **busnum = 1**

8. set up the software to start at boot, using commands similar to the following by changing where the scripts are stored (if not in /root):

```
sudo 'echo sudo python /root/
  lift_lights.py>>/etc/rc.local'
sudo 'echo sudo python /root/
  lift-lights-with-espeak.py>>
  /etc/rc.local'
```

Bera Somnath,
Vindhyanagar, India. ($95)

Circuit Ideas Wanted

Got an interesting original circuit that you have cleverly devised? We need it and will pay good money to feature it in the Circuit Notebook pages. We can pay you by electronic funds transfer, cheque (what are they?) or direct to your PayPal account. Or you can use the funds to purchase anything from the SILICON CHIP on-line shop, including PCBs and components, back issues, subscriptions or whatever. Email your circuit and descriptive text to editor@siliconchip.com.au