PiDP-11/70 Test Program

Table of Contents

Revision History

Date	Who	Version	Description
20180804	Neil Higgins	0.1DRAFT	Initial draft
20180807	Neil Higgins	0.2DRAFT	Treatment of rotary encoders corrected
20180808	Neil Higgins	0.3DRAFT	Version raised to match implementation
20180812	Neil Higgins	0.4DRAFT	Version raised to match implementation
20180813	Neil Higgins	0.5DRAFT	Selective inversion of switches added, to enable display of logical state
20180909	Neil Higgins	0.6DRAFT	Figure 3 updated
20181013	Neil Higgins	0.7DRAFT	Functions section updated
20181017	Neil Higgins	0.8DRAFT	Updates suggested by Oscar: • Code - Check Python version • Code - Check to see if server running • Doc - Fix schematic

Licence

Free as (in beer) and free (as in thought): Use at your own risk.

Functions

1. Step through all LEDs on each electronic row (led0 - led5)
2. Step through all LEDs on each physical row
3. Display switches and switch deltas, in octal, until **CTRL+C** is typed
4. Display rotary encoders, modulo 10, until **CTRL+C** is typed

Connections (from schematic)

			led0	led1	led2	led3	led4	led5	row0	row1	row2
		Pin	38	40	15	16	18	22	36	11	12
		GPIO	20	21	22	23	24	25	16	17	18
	Pin	GPIO									
col0	37	26	LED16 A0	LED4 A12	LED32 ADD22	LED48 D0	LED36 D12	-	SW7 SR0	SW29 SR12	SW28 TEST
col1	13	27	LED15 A1	LED3 A13	LED31 ADD18	LED47 D1	LED35 D13	-	SW8 SR1	SW30 SR13	SW21 L-ADD
col2	7	4	LED14 A2	LED2 A14	LED30 ADD16	LED46 D2	LED34 D14	-	SW9 SR2	SW31 SR14	SW22 EXAM
col3	29	5	LED13 A3	LED1 A15	LED29 DATA	LED45 D3	LED33 D15	-	SW10 SR3	SW32 SR15	SW23 DEP
col4	31	6	LED12 A4	LED22 A16	LED28 KRNL	LED44 D4	LED50 PAR-L	-	SW11 SR4	SW1 SR16	SW24 CONT
col5	26	7	LED11 A5	LED21 A17	LED27 SUPER	LED43 D5	LED49 PAR-H	-	SW12 SR5	SW2 SR17	SW25 ENA
col6	24	8	LED10 A6	LED20 A18	LED26 USER	LED42 D6	LED59 R1-U-D	LED60 R1-U-I	SW13 SR6	SW3 SR18	SW26 SING_I
col7	21	9	LED9 A7	LED19 A19	LED25 MASTR	LED41 D7	LED58 R1-S-D	LED61 R1-S-I	SW14 SR7	SW4 SR19	SW19 START
col8	19	10	LED8 A8	LED18 A20	LED24 PAUSE	LED40 D8	LED57 R1-K-D	LED62 R1-K-I	SW15 SR8	SW5 SR20	SW34 E1A
col9	23	11	LED7 A9	LED17 A21	LED23 RUN	LED39 D9	LED64 R1-C-P	LED63 R1-P-P	SW16 SR9	SW6 SR21	SW35 E1B
col10	32	12	LED6 A10	-	LED52 A_ERR	LED38 D10	LED56 R2-D-P	LED53 MU-A	SW17 SR10	SW33 E1 ADDR	SW37 E2A
col11	33	13	LED5 A11	-	LED51 P_ERR	LED37 D11	LED55 R2-B-R	LED54 R2-D-R	SW18 SR11	SW36 E2 DATA	SW38 E2B

Notes:

1. There is no SW20 or SW27
2. Key switch is assumed not connected, and ignored

Physical rows (top to bottom, left to right)

LEDs

1. LED51, LED52, LED23, LED24, LED25, LED26, LED27, LED28, LED29, LED30, LED31, LED32, LED59, LED60
2. LED58, LED61
3. LED57, LED62
4. LED64, LED63
5. LED17, LED18, LED19, LED20, LED21, LED22, LED1, LED2, LED3, LED4, LED5, LED6, LED7, LED8, LED9, LED10, LED11, LED12, LED13, LED14, LED15, LED16
6. LED56, LED53
7. LED55, LED54
8. LED49, LED50, LED33, LED34, LED35, LED36, LED37, LED38, LED39, LED40, LED41, LED42, LED43, LED44, LED45, LED46, LED47, LED48

Switches

1. SW33/SW34/SW35 (E1)
2. SW36/SW37/SW38 (E2)
3. SW6, SW5, SW4, SW3, SW2, SW1, SW32, SW31, SW30, SW29, SW18, SW17, SW16, SW15, SW14, SW13, SW12, SW11, SW10, SW9, SW8, SW7, SW28, SW21, SW22, SW23, SW24, SW25, SW26, SW19

Abridged Schematic

See Figure 1.

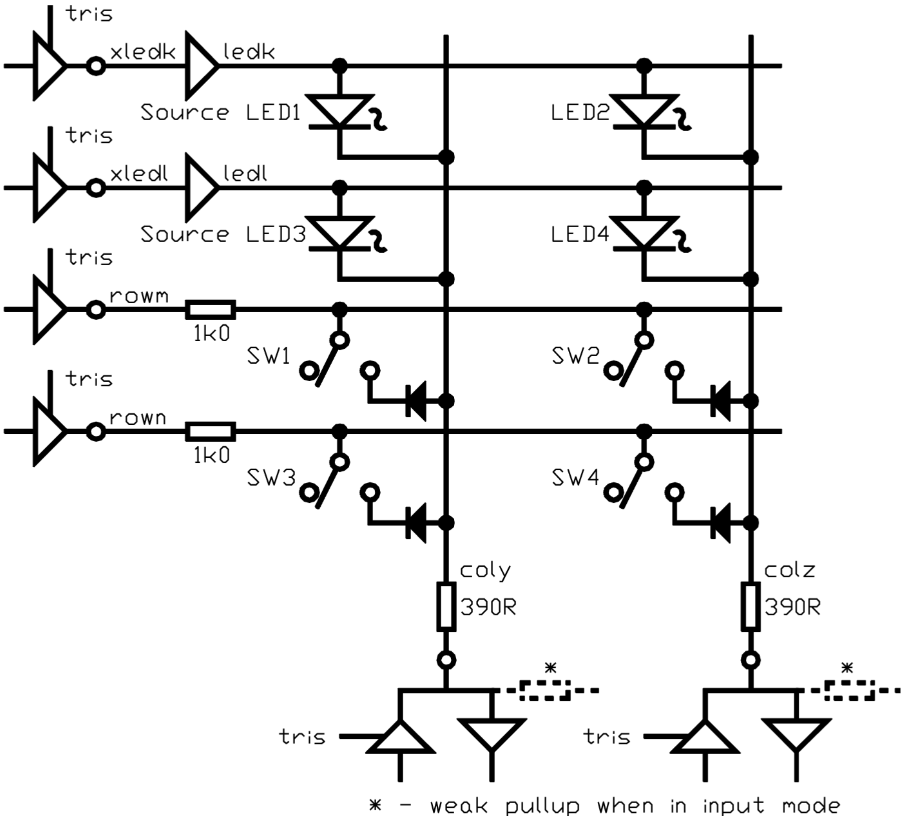


Figure 1

Driving LEDs and Reading Switches

Note: These are the basics - numerous speed-ups are possible.

LEDs

Default:

- All (x)led and columns tristate

LED On: Set (x)led high and column low
LED Off: Set (x)led low and column low

Switches

Default:

- All rows and columns tristate

Read switch:

- Set column to input with weak pull-up, set row low, read column

Rotary Encoders

The rotary encoders are modulo 4 Gray code encoders which rely on rubbing contact to provide reliable indication at a low price. The common terminal of the A and B switches of each encoder must be connected to a row.

Each rotary encoder reads identically in all detent positions. The A and B switches undergo a complete cycle between detent positions. Because of this, the switches must be sampled frequently and continuously. The detent state is A open & B open. Depending on the direction of rotation, one of two sequences will be detected: A open & B closed; A closed & B closed, A closed & B open; returning to A open & B open; or A closed & B open; A closed & B closed; A open & B closed; returning to A open & B open. See Figure 2.

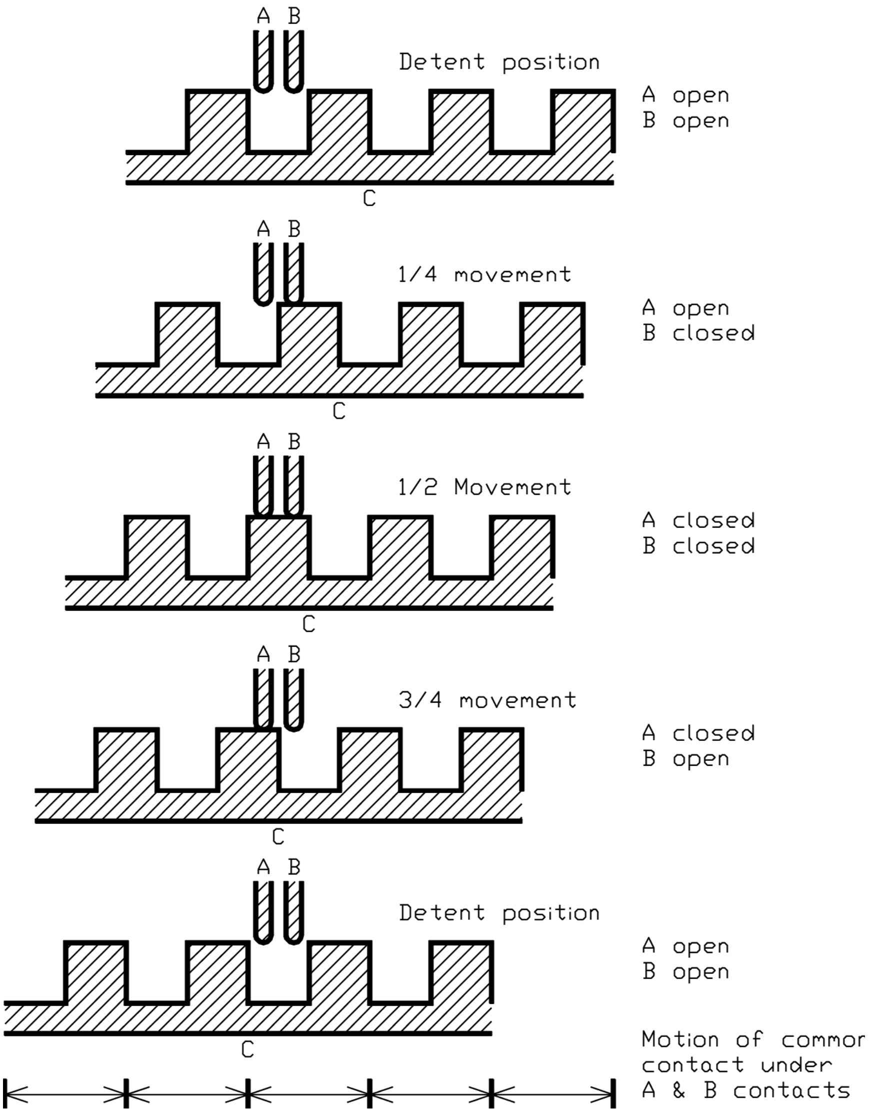


Figure 2

Configuration Data

- A tuple per LED: LED number, LED name, LED function, LED electronic row GPIO, LED electronic column GPIO, LED physical row, LED physical column
- A tuple per switch: SW number, SW name, SW function, SW electronic row GPIO, SW electronic column GPIO, SW invert flag, SW physical row, SW physical column, Octal digit, Octal digit weight
- A tuple per Octal digit: DIG number, DIG name, DIG function, Display row, Display column

All within the Python program.

Implementation

See PiDP1170Test.py

Operation

1. Shut down the PiDP-11/70 client (simh) and server (server11)
   a) At the PDP-11 operating system prompt, type **CTRL+E** to get the simh> prompt
   b) Type **exit** (followed by Enter) to shut down simh
   c) Experience has shown that this also shuts down the server process (mechanism unknown)
   d) At the Linux prompt, type **ps ax** (followed by Enter) and check the process list. There should be no instances of pidp11.sh or server11 running. Equivalently **ps ax | grep pidp** and **ps ax | grep server** should find no matches except the command line

2. Run the test program, PiDP1170Test.py
   a) Open, and for best display of results maximise, a command window
   b) At the Linux prompt, type **python3 PiDP1170Test.py** (followed by Enter)
   c) The test program cycles the LEDs by row and column GPIOs, and then by physical rows and columns
   d) Type Enter to proceed to switch testing
   e) The test program loops once per second, reading and displaying the switches. The switch register switches are displayed in octal; the others, including the press switches in the rotary encoders, are displayed in binary
   f) Type **CTRL+C** to stop switch testing
   g) Type Enter to proceed to rotary encoder testing
   h) Each rotary encoder is displayed as a number (modulo 10) which increases as the shaft is rotated in one direction and decreases as the shaft is rotated in the opposite direction
   i) Type **CTRL+C** to terminate the program

Anomalous Conditions

If you try to start the program with **python PiDP1170Test.py** (i.e. Python 2), rather than **python3 PiDP1170Test.py**, the program will announce that it requires Python 3, then exit.

When you start the program, it may announce that “It looks as if the blinkenlight server is, or has been, running”. This is detected by examining the GPIO configuration.

- If you are sure that the server is not running, press the Enter key to continue. You may subsequently see one or more warnings, but these can be ignored
- Otherwise type **CTRL+C** to terminate the program, then find the pid of the server program and kill it

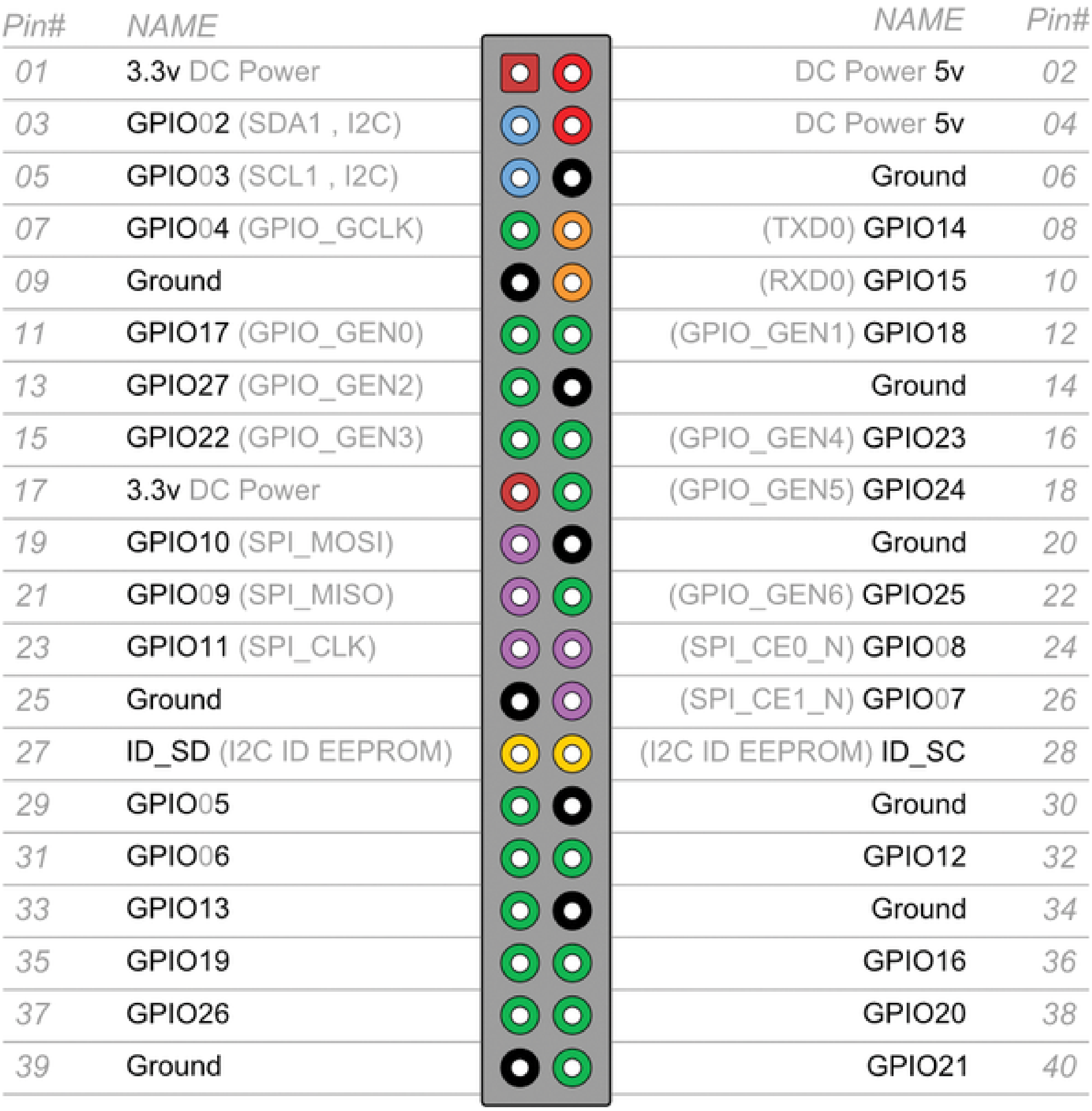
References

1. PiDP-11 web site - http://obsolescence.wixsite.com/obsolescence/pidp-11
2. PiDP-11 Google Group - https://groups.google.com/forum/#!forum/pidp-11
3. Python language - https://www.python.org/
4. RPi.GPIO library - https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/

GPIO-to-pin mapping for Raspberry Pi

See Figure 3.

Source: http://www.linux-magazine.com/Online/Features/Raspberry-Pi-Model-B/(offset)/3



Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Figure 3