ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Kerberos: DSL-Configurable burglar alarm system for the Raspberry Pi

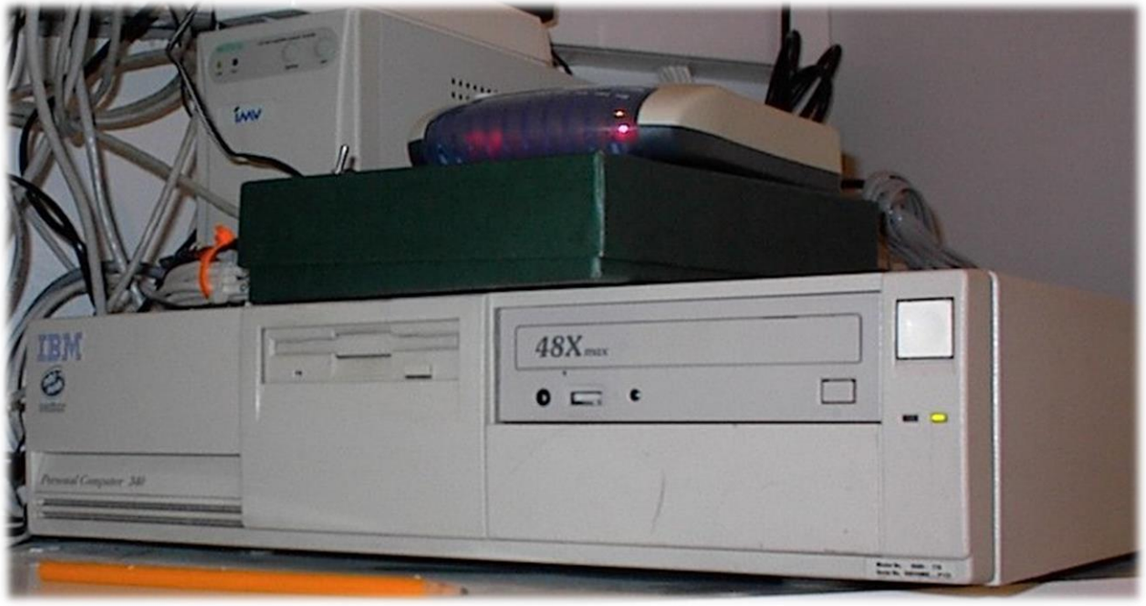
Diomidis Spinellis
Professor, Department of Management Science and Technology
http://www.dmst.aueb.gr/dds
@CoolSWEng

Existing Home Alarms

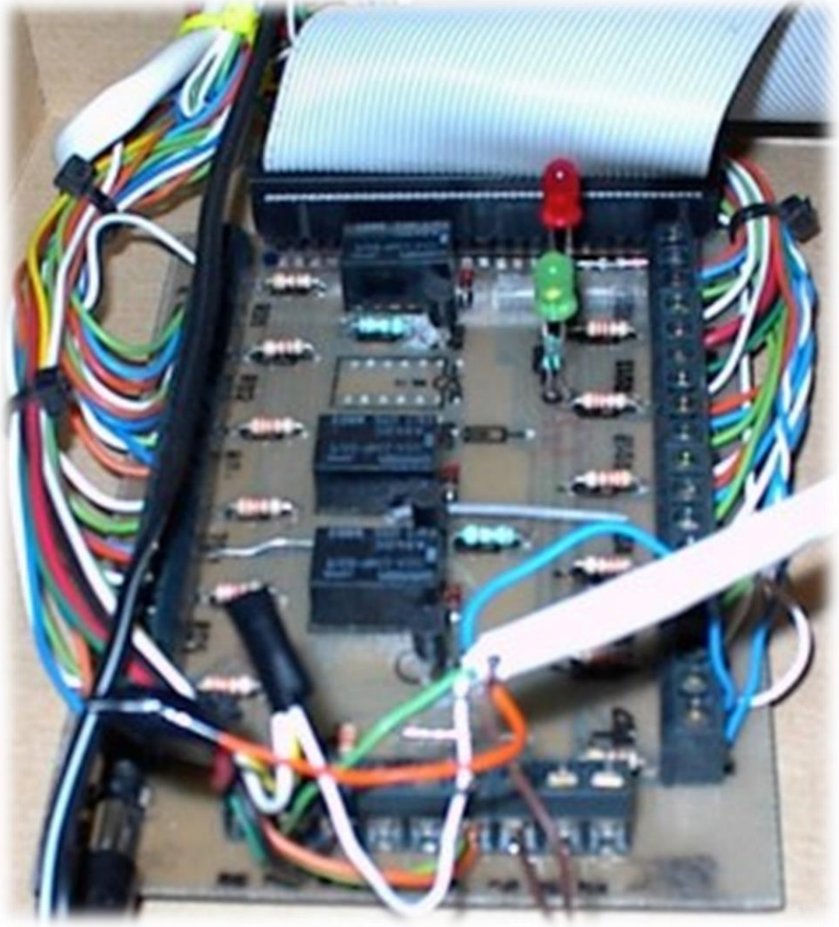
Existing Home Alarms

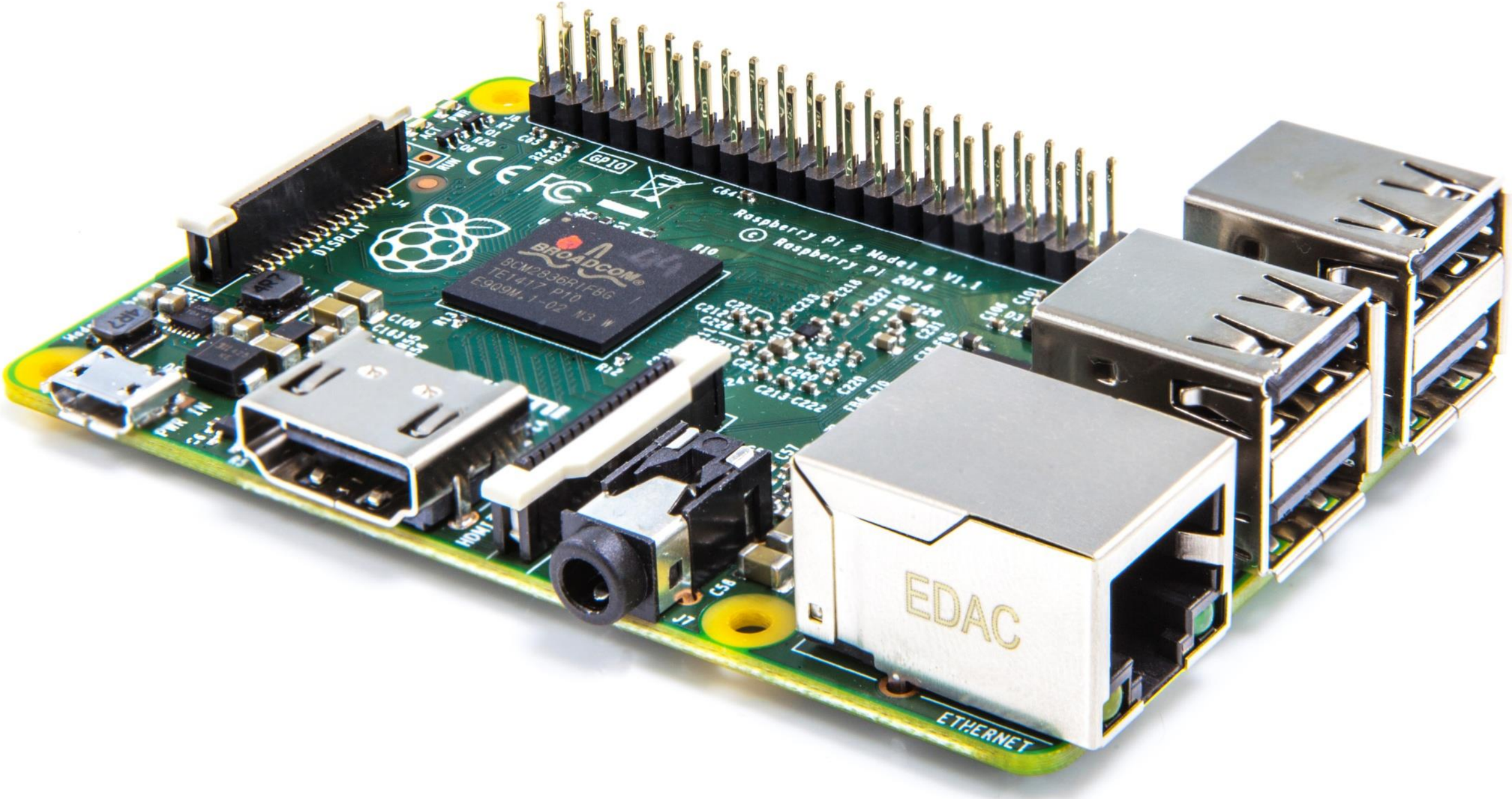
- Vendor tie up
- Backdoors, known vulnerabilities
- Poor home automation integration
- Simplistic features, UI
- Not user-hackable
- NIH

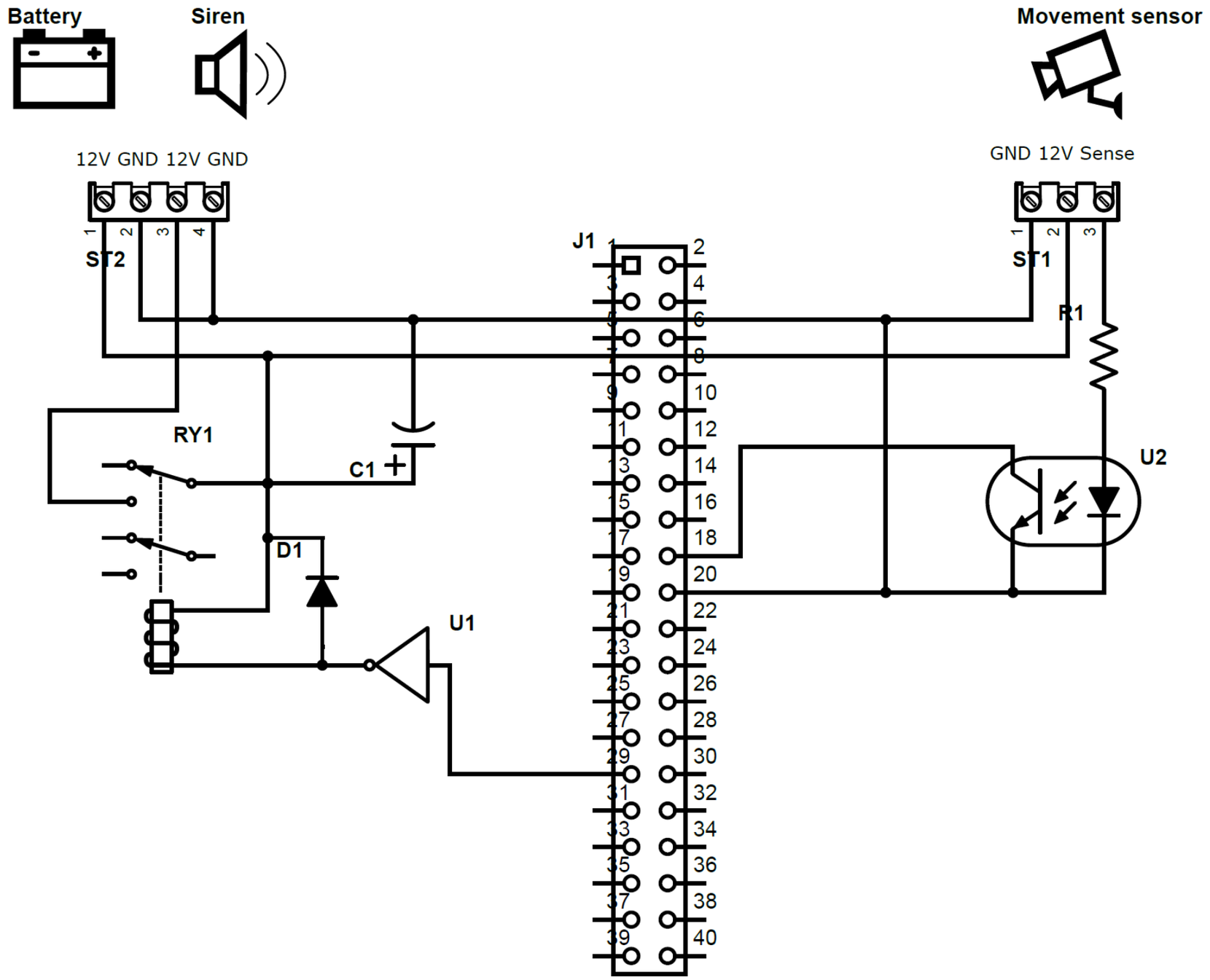

Tamper
Unset
Power
TO FULL SET
Enter your code. Press FULL.
TO PART SET
Enter your code. Press part set number. Press PART.
TO UNSET
Enter your code
TO SILENCE THE ALARM
Enter your code. Press RESET
Zone 1
Zone 2
Zone 3
Zone 4
Zone 5
Zone 6
Zone 7
Zone 8
1
2
3
OMIT
4
5
6
7
8
9
0
FULL
PART
PROG.
RESET


IBM
48X max
Personal Computer 300


ISO 9001 14001 Certified
PCL-724 24 BIT DIGITAL I/O CARD REV.A1
A8A022
Automation with PCs
ADVANTECH
PC-LabCard SERIES

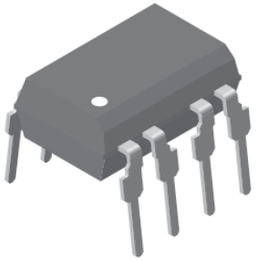



Raspberry Pi 2 Model B V1.1
© Raspberry Pi 2014
BROADCOM
BCM2836RIFBG
TE1417 PTO
E9Q9M.1-02 N3 W
DISPLAY
HDMI
PWR IN
EDAC
ETHERNET

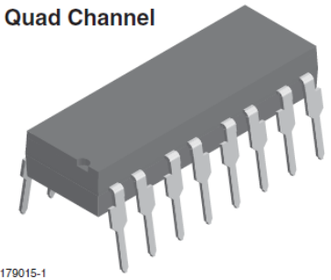

Battery
Siren
Movement sensor
12V GND 12V GND
GND 12V Sense
ST2
ST1
J1
R1
RY1
C1
D1
U1
U2

Optocoupler, Phototransistor Output (Dual, Quad Channel)

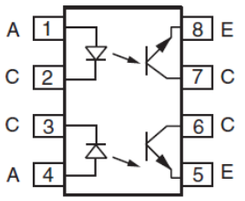
Dual Channel

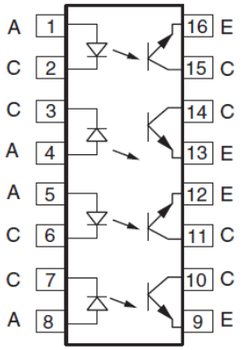


Quad Channel



i179015-1





FEATURES

- ILD74, ILQ74 TTL compatible
- Transfer ratio, 35 % typical
- Coupling capacitance, 0.5 pF
- Single, dual, and quad channel
- Industry standard DIP packages
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Pb-free

e3

RoHS COMPLIANT

AGENCY APPROVALS

- UL1577, file no. E52744 system code H or J, double protection
- CSA 93751
- BSI IEC 60950; IEC 60065
- DIN EN 60747-5-2 (VDE 0884)/DIN EN 60747-5-5 pending available with option 1
- FIMKO

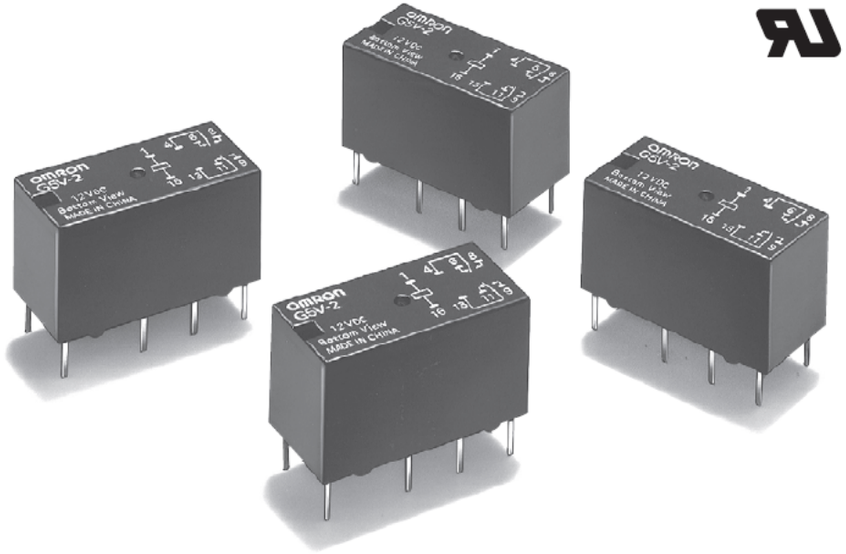
DESCRIPTION

The ILD74, ILQ74 is an optically coupled pair with a GaAlAs infrared LED and a silicon NPN phototransistor. Signal information, including a DC level, can be transmitted by the device while maintaining a high degree of electrical isolation between input and output.

G5V-2

Low Signal Relay

General-purpose, Low-cost, Two-pole Relays for Signal Circuits



- General-purpose DIL terminal layout.
- Wide switching power of 10 µA to 2 A.
- Fully-sealed type Relays standardized with bifurcated crossbar contacts. Highly reliable in addition to its high environment resistance.
- Conforms to FCC Part 68 (impulse withstand voltage of 1,500 V for 10 x 160 µs between coil and contacts and between contacts of the same polarity).
- High dielectric strength at 1,000 VAC between coil and contacts, and 750 VAC between contacts of the same polarity.
- UL and CSA standard approved.

RoHS Compliant

■Model Number Legend

G5V-□-□
1 2

1. Number of Poles/ Contact form

2: 2-pole/DPDT (2c)

2. Classification

None: Standard
H1: High-sensitivity

■Application Examples

- Telecommunication equipment
- Security equipment

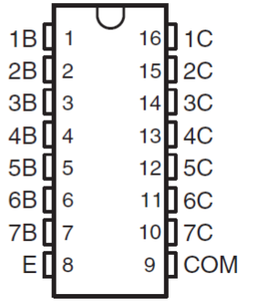
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON TRANSISTOR ARRAYS

Check for Samples: ULN2002A, ULN2003A, ULN2003AI, ULN2004A, ULQ2003A, ULQ2004A

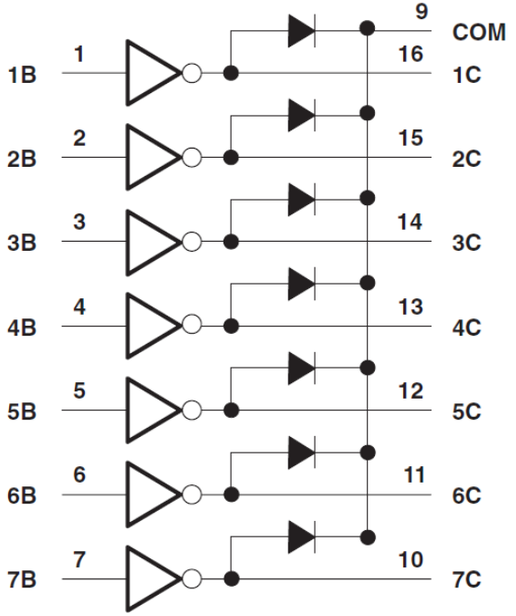
FEATURES

- 500-mA-Rated Collector Current (Single Output)
- High-Voltage Outputs: 50 V
- Output Clamp Diodes
- Inputs Compatible With Various Types of Logic
- Relay-Driver Applications

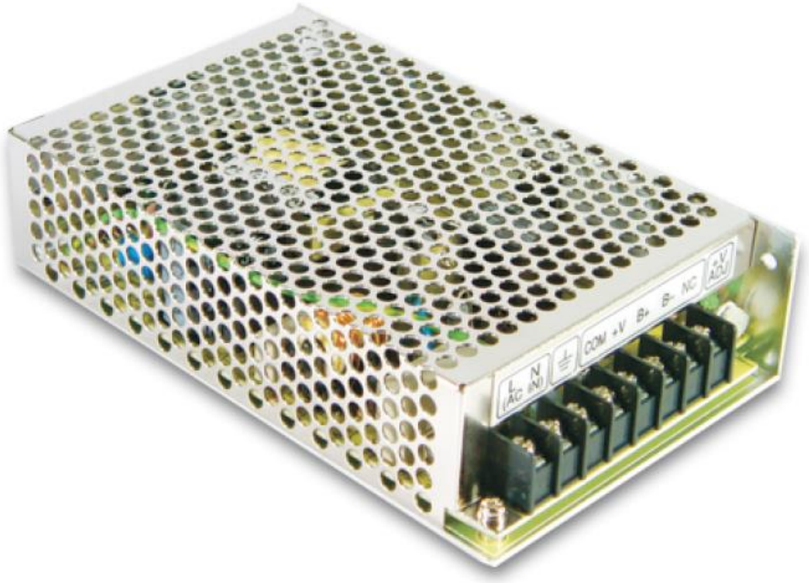
ULN2002A . . . N PACKAGE
ULN2003A . . . D, N, NS, OR PW PACKAGE
ULN2004A . . . D, N, OR NS PACKAGE
ULQ2003A, ULQ2004A . . . D OR N PACKAGE
(TOP VIEW)



LOGIC DIAGRAM

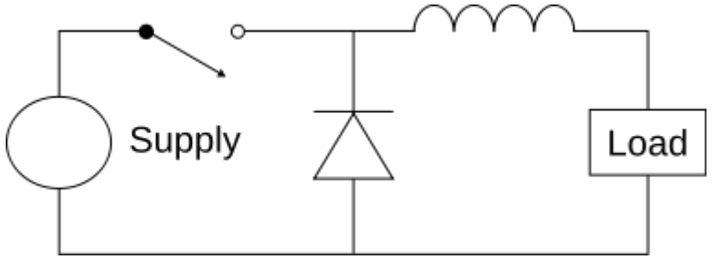


55W Single Output with Battery Charger(UPS Function) AD-55 series

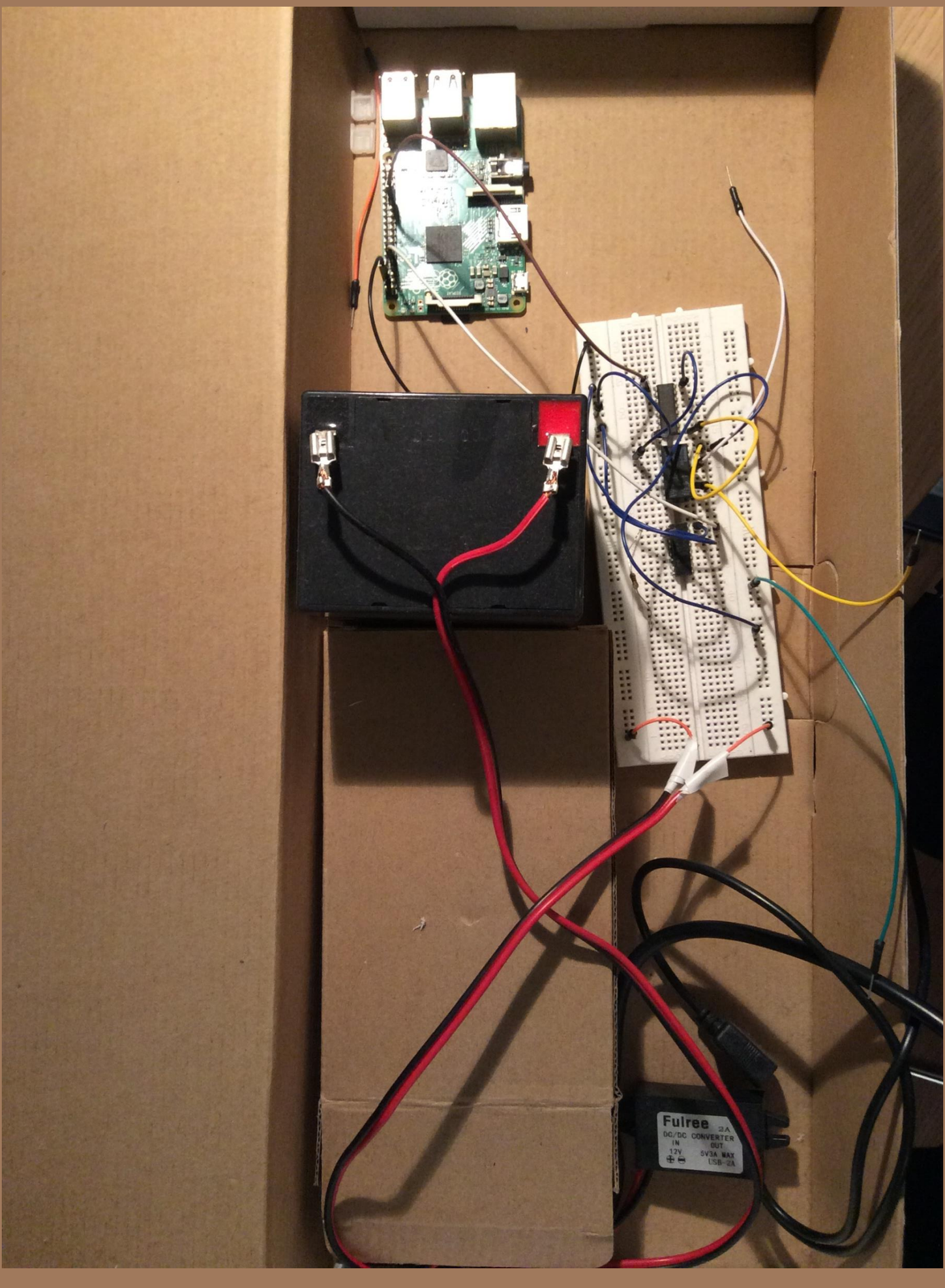


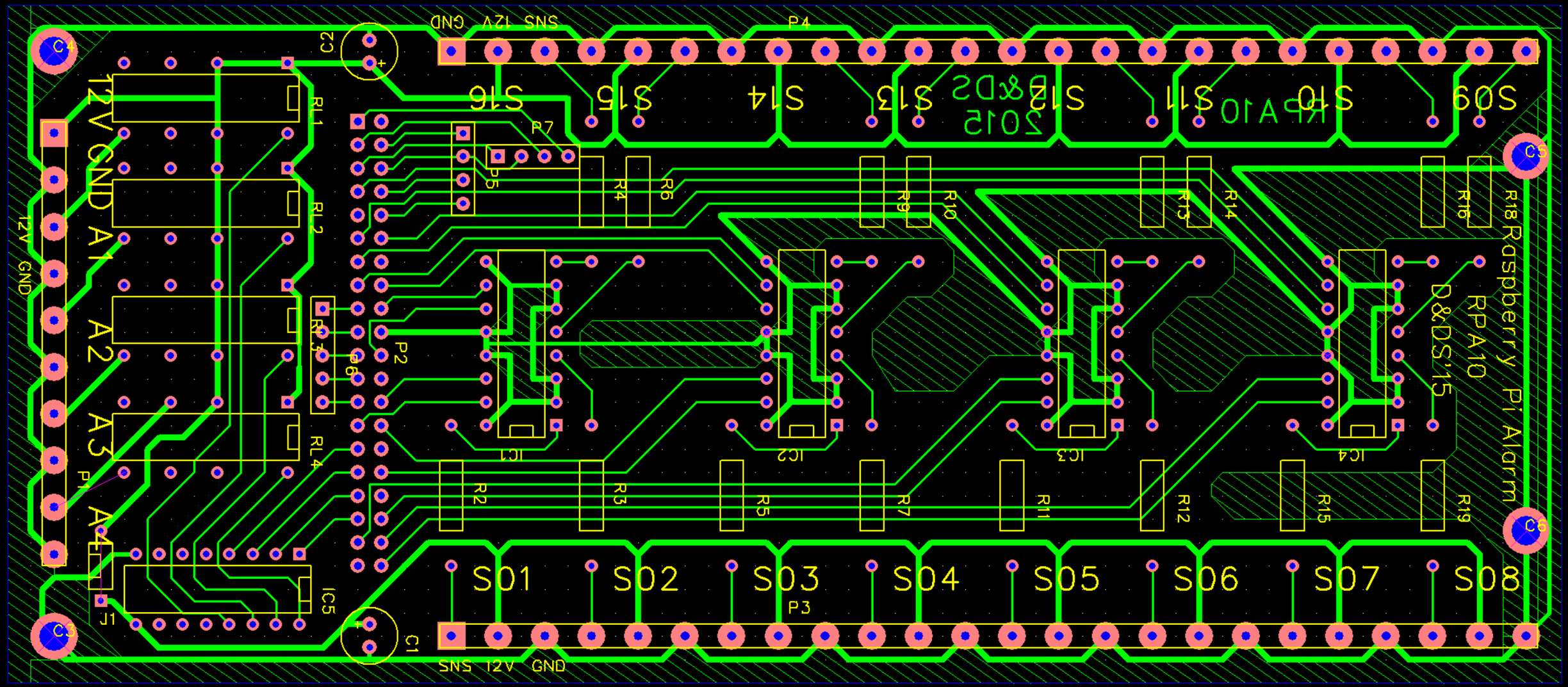
■ Features :

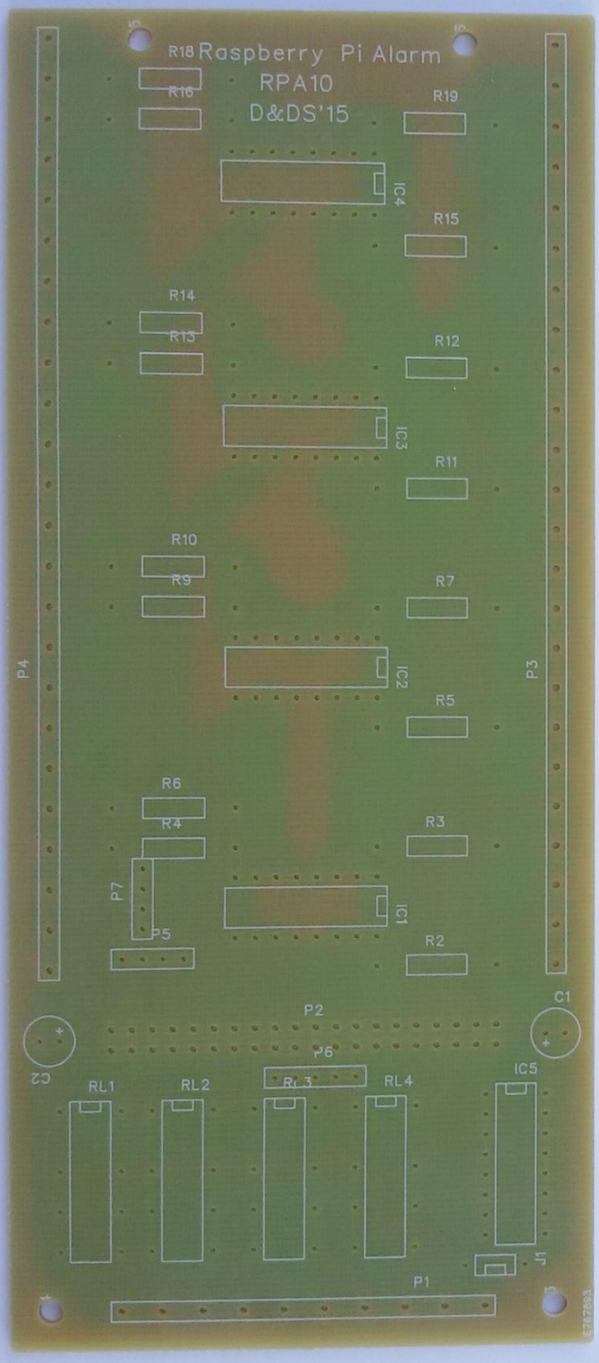
- Universal AC input / Full range
- Protections: Short circuit / Overload / Over voltage
- Battery low protections
- Cooling by free air convection
- 100% full load burn-in test
- Fixed switching frequency at 45KHz
- 2 years warranty

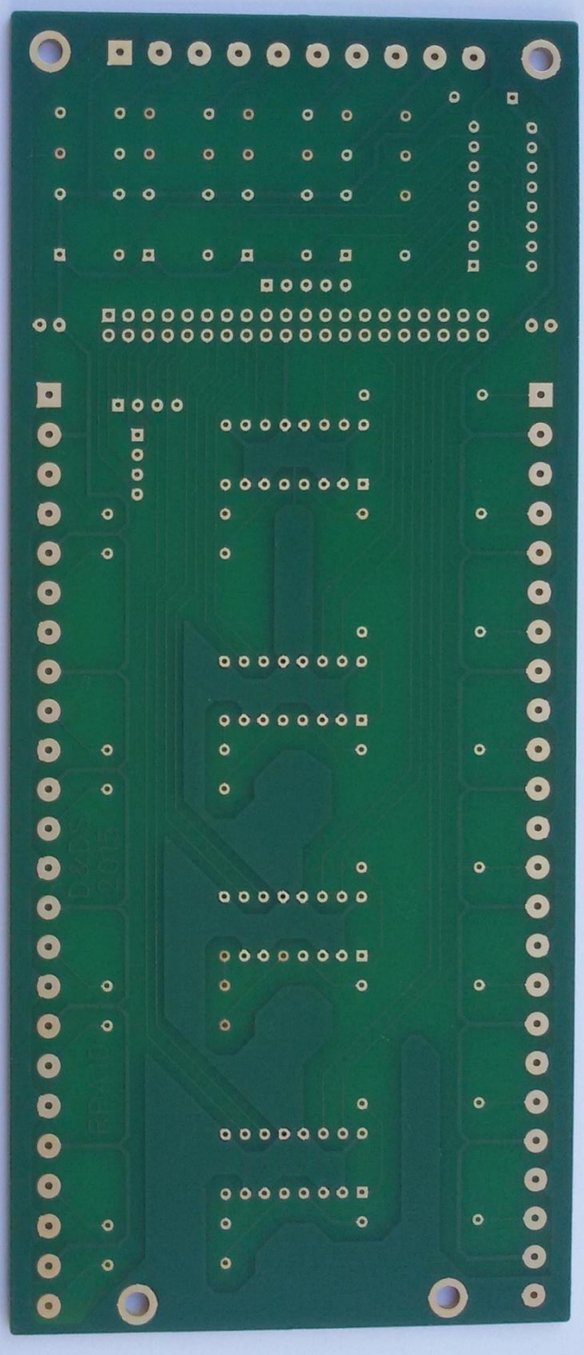

Supply
Load

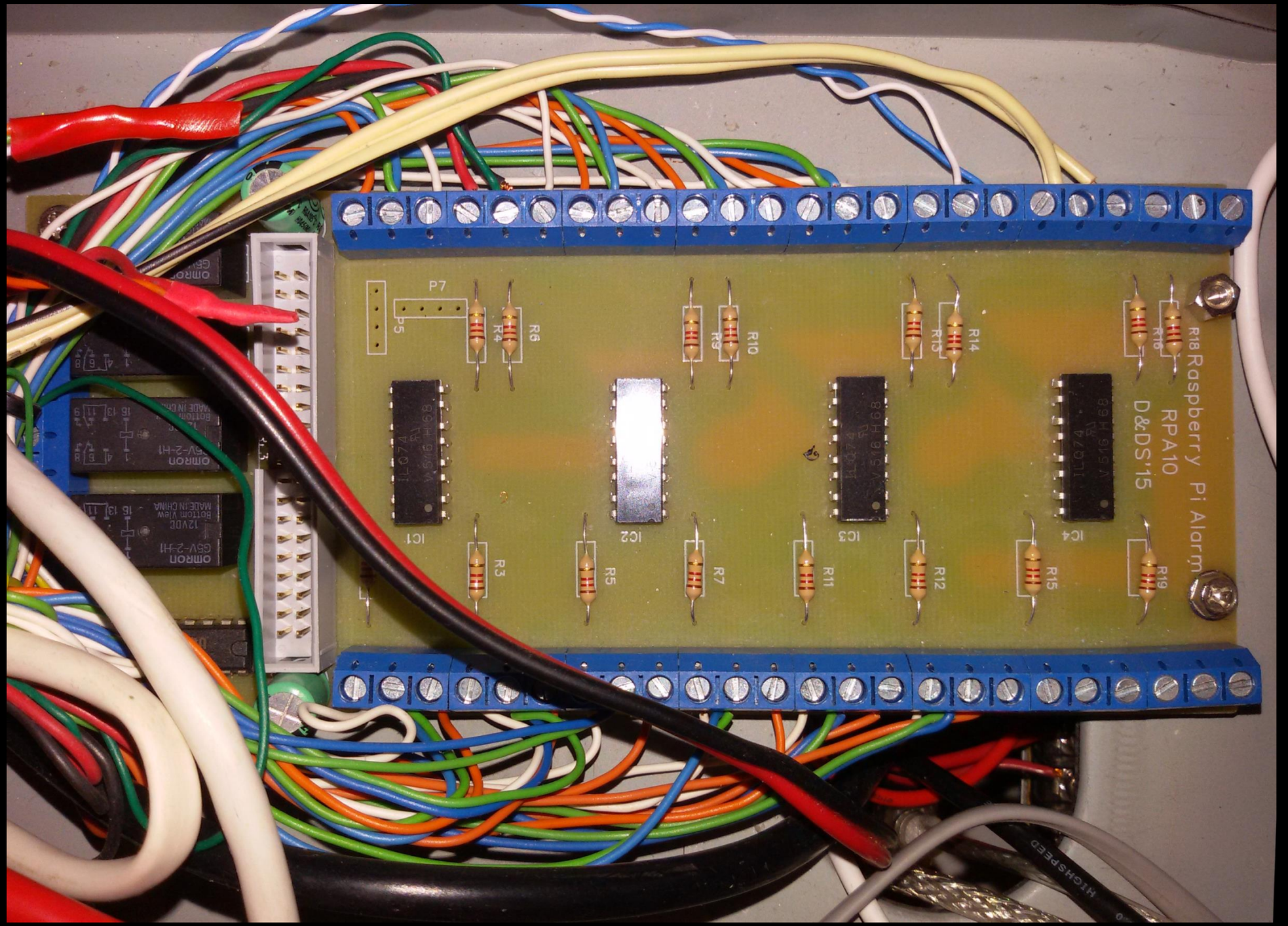

CPT
012050-3
15W DC/DC CONVERTER
IN(綠)
OUT(藍)
R(紅)
Y(黃)
B(黑)
B(黑)
12V
5V/3A MAX

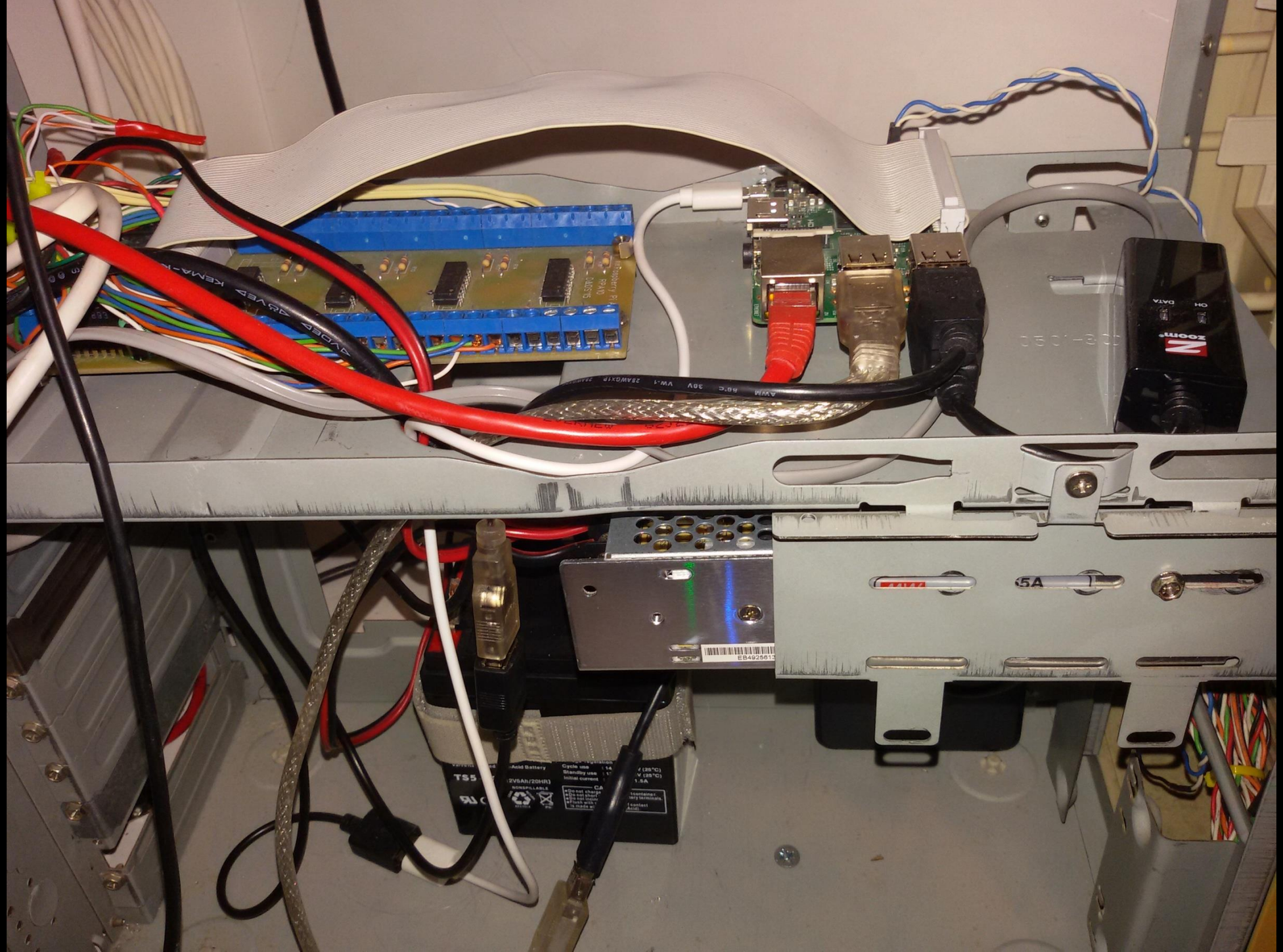

Fulree
2A
DC/DC CONVERTER
IN
12V
OUT
5V3A MAX
USB-2A


Raspberry Pi Alarm
RPA10
D&DS'15
12V GND A1 A2 A3 A4
S01 S02 S03 S04 S05 S06 S07 S08
S09 S10 S11 S12 S13 S14 S15 S16
SNS 12V GND
P1 P2 P3 P4 P5 P6 P7
RL1 RL2 RL4
IC1 IC2 IC3 IC4 IC5
R2 R3 R5 R7 R11 R12 R15 R19
R4 R6 R9 R10 R13 R14 R16 R18
C1 C2 C3 C4 C5
J1


Raspberry Pi Alarm
RPA10
D&DS'15
R18
R16
R19
IC4
R15
R14
R13
R12
IC3
R11
R10
R9
R7
IC2
R5
R6
R4
R3
P7
IC1
P5
R2
P4
P3
P2
C1
C2
P6
RL1
RL2
RL3
RL4
IC5
J1
P1



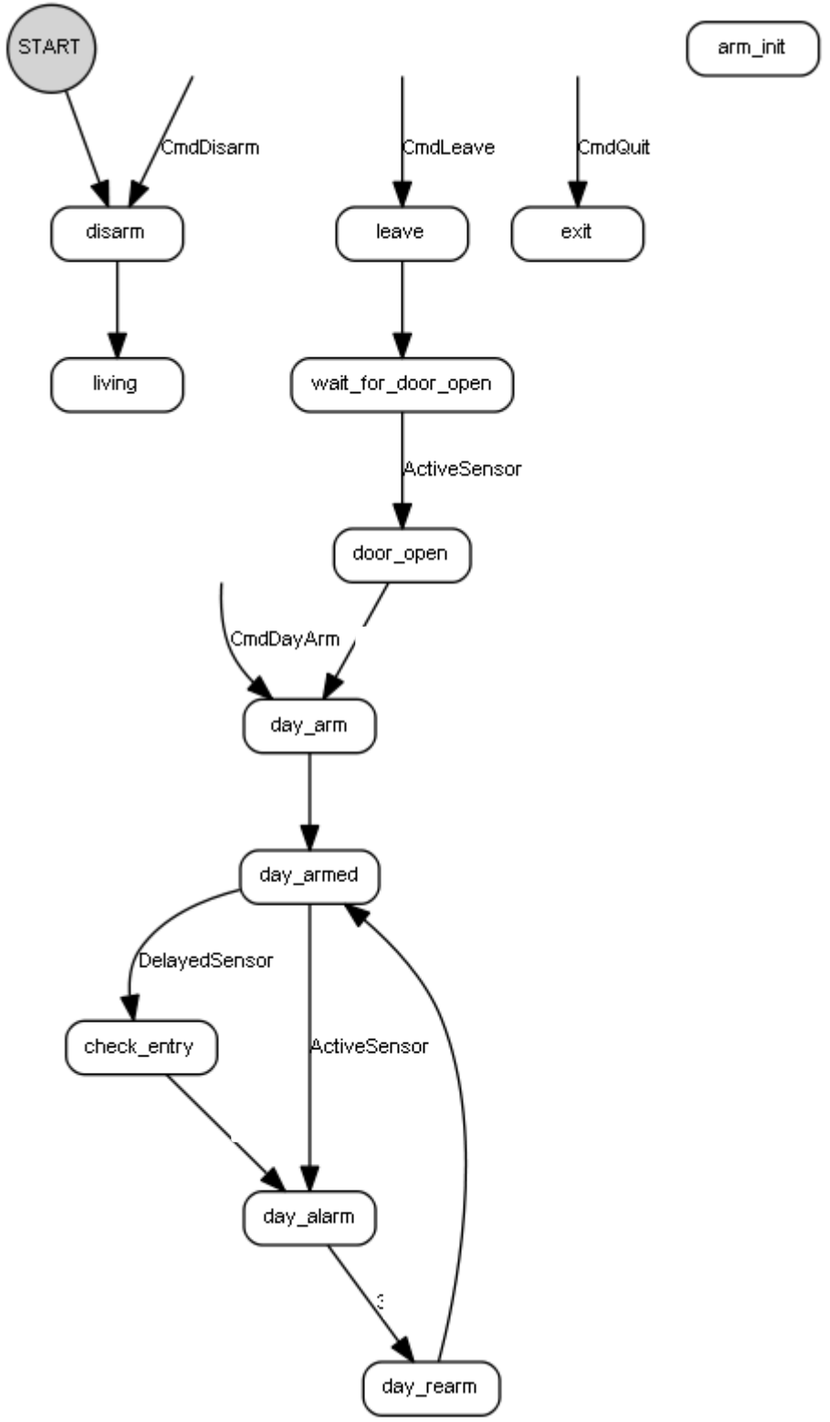

Raspberry Pi Alarm
RPA10
D&DS'15
P7
P5
R4
R6
R3
R10
R5
R14
R7
R11
R12
R15
R16
R19
R18
IC1
IC2
IC3
IC4
OMRON
G5V-2-H1
12VDC
Bottom View
MADE IN CHINA
HIGHSPEED


zoom
CH
DATA
5A

```
%{
#include <stdlib.h>
#include <syslog.h>
#include <unistd.h>
#include "evlst.h"
#include "alarm.h«
%}

%i disarm

*:
      CmdDayArm > day_arm
      CmdLeave > leave
      CmdDisarm > disarm
      CmdQuit > exit
      ;

exit:
      | set_bit("Siren", MAIN_OFF)
      | closelog()
      | exit(0)
      ;

disarm:
      | syslog(LOG_INFO, "Disarmed")
      | set_bit("Siren", MAIN_OFF)
      | unlink(STATUSPATH "dayarm")
      | touch(STATUSPATH "disarm")
      | set_sensor_active(ALL, OFF)
      | zero_sensors()
      > living
      ;

leave:
      | set_sensor_active(ALL, OFF)
      | set_sensor_active("Door2", ACTIVE)
      > wait_for_door_open
      ;

wait_for_door_open:
      | syslog(LOG_INFO, "Waiting for door open")
      ActiveSensor > door_open
      ;

door_open:
      | syslog(LOG_INFO, "Opening outside door")
      | system("vm shell -v -x 1 -l modem -S /usr/bin/perl " SCRIPTDIR
      "/call.pl 32")
         ; > day_arm
      ;

arm_init:
      | zero_sensors()
      | set_sensor_active(ALL, ACTIVE)
      | ClearCounter(day_alarm)
      | unlink(STATUSPATH "dayarm")
      | unlink(STATUSPATH "disarm")
      ;

day_arm:
      | call arm_init
      | set_sensor_active("Door", DELAYED)
      | touch(STATUSPATH "dayarm")
      > day_armed
      ;

day_armed:
      | syslog(LOG_INFO, "Day armed")
      DelayedSensor > check_entry
      ActiveSensor > day_alarm
      ;

check_entry:
         > day_alarm
      ;

day_alarm:
      | syslog(LOG_INFO, "Day alarm")
      | increment_sensors()
      | set_bit("Siren", MAIN_ON)
         ; > day_rearm
      ;

day_rearm:
```

Possible Features

- Front ends: voice, web, SMS
- Arm on leave
- Forgot something
- Auto night arm
- Night movements
- Visitor entry
- Turn on lights
- Warning on entry
- Auto disarm
- Camera integration


START
arm_init
CmdDisarm
CmdLeave
CmdQuit
disarm
leave
exit
living
wait_for_door_open
ActiveSensor
door_open
CmdDayArm
day_arm
day_armed
DelayedSensor
check_entry
ActiveSensor
day_alarm
day_rearm

```
static void
proc_ST_check_entry(void)
{
   state_count[ST_check_entry]++;
   syslog(LOG_DEBUG, "state: ST_check_entry (%d)",
   state_count[ST_check_entry]);
   register_timer_event(5, EV_TIMER_5);


   for (;;) switch(get_event()) {
   case EV_CmdDayArm: state = ST_day_arm; return;
   case EV_CmdLeave: state = ST_leave; return;
   case EV_CmdDisarm: state = ST_disarm; return;
   case EV_CmdQuit: state = ST_exit; return;

   case EV_TIMER_5: state = ST_day_alarm; return;

   }
}
```

Metrics

- DSL file: 238 lines
- Other C code: 374 lines
- Compiler DSL to C: 128 lines
- Visualizer DSL to dot: 66 lines
- Generated C code: 688 lines

Lessons Learned

- DSL flexibility
- Need for (difficult) testing
- Need for maintenance, upgrades
- DSL survived complete upgrade
- Difficult for non-experts

Thank you!

www.spinellis.gr

@CoolSWEng