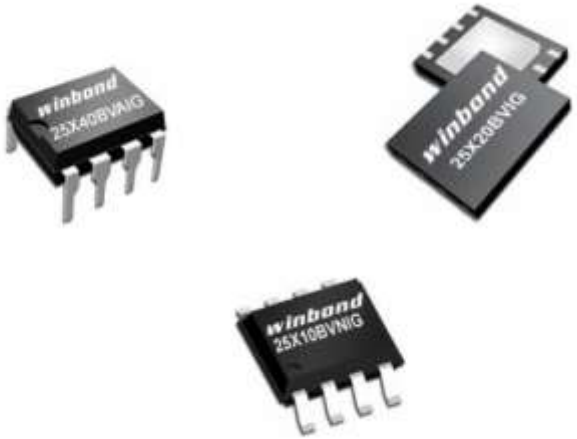
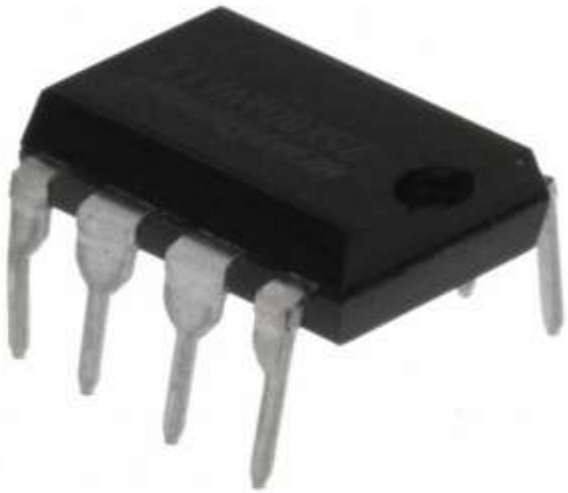
# Raspberry Pi Flash Programmer




Nick Navarro

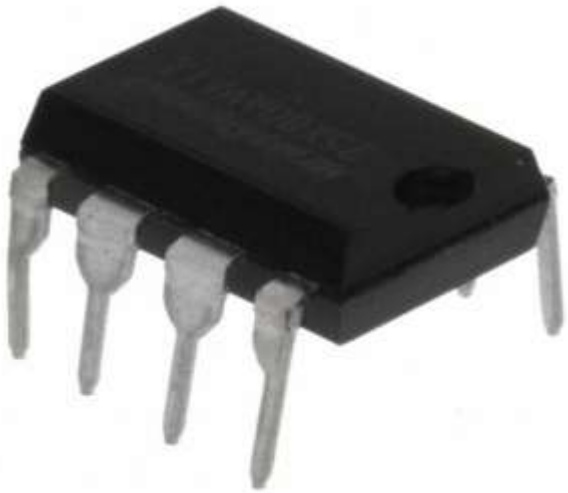
# Flashrom

- flashrom can do stuff to flash chips
  - Identifies, Reads, Writes, Verifies, Erases
- Flashes mainboards/NICs/storage cards,etc.
  - BIOS
  - EFI
  - Coreboot
  - Custom Firmware
  - Option ROM images
- Supports LPC, FWH, SPI, SOIC8, Etc.
- Hundreds of flash chips and dozens of motherboard chipsets



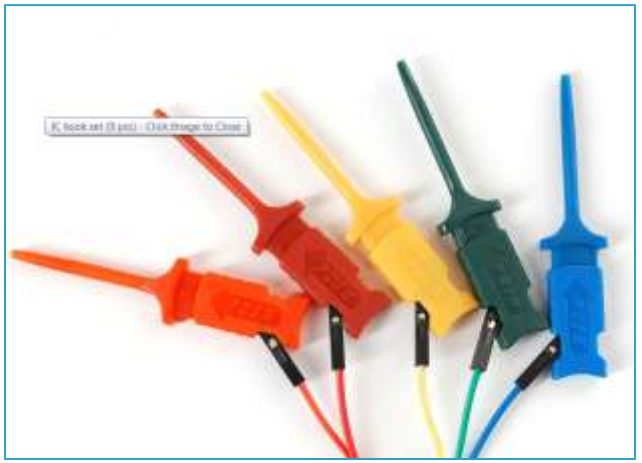
# Serial Peripheral Interface (SPI)

- They are serial, contrary to parallel
  - Simpler wiring
  - Serial buses are fast enough
  - Examples of serial: USB $I^2C$, CAN bus etc.
- SPI flash EEPROM has replaced larger PLCC and TSOP EEPROM
- Can be programmed without removing the EPROM
- Easier to debug than PLCC
- Often uses the SOIC or small-outline integrated circuit for surface mounting



# In-System Programming (ISP)

- The ability to write to a flash attached to the circuit
- Easier than removing the flash chip and socketing it
- Usually done with SPI chips only which is my target for this experiment
- Sometimes the SPI bus is not isolated enough and parts of the chipset have partial power
- Can use 8-pin SOIC clip
- Can also use the IC hook clips

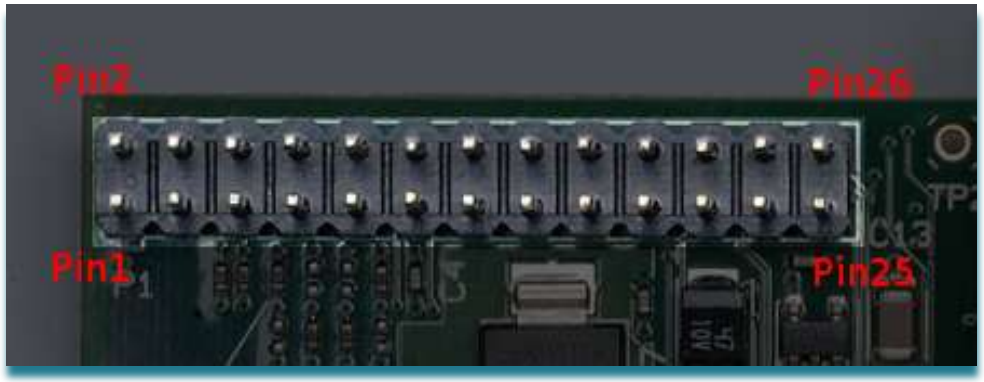


# Raspberry Pi

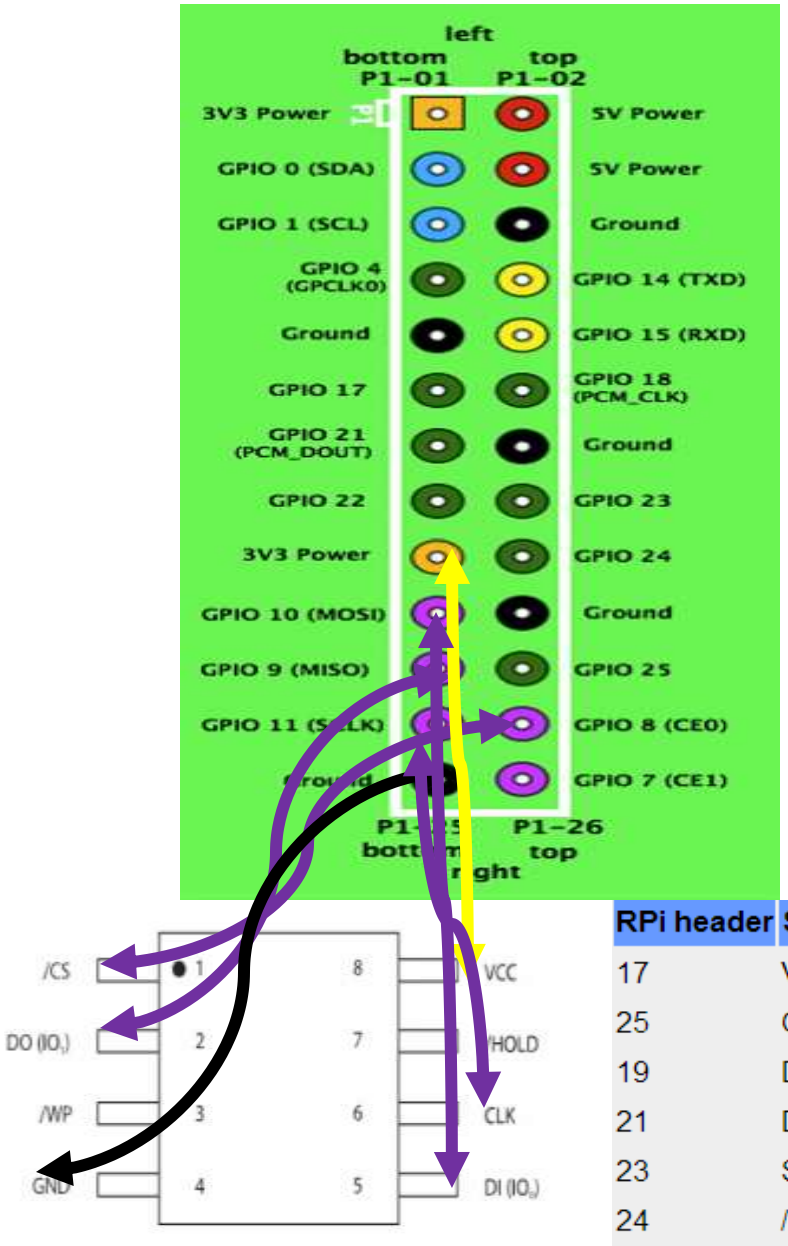
- $25 or $35 single-board computer
- Has an ARM1176JZF-S 700 Mhz processor
- Broadcom BCM2835 built-in NIC
- SD card for storage
- Runs Raspbian (Debian) Linux and BSD
- Good for electronics projects (see Adafruit)
  - Power control, sensing movement, controlling motors, temperature sensing, motion sensing

# General Purpose Input/Output (GPIO)

- Generic pin on a chip that can be programmed
- Allows peripherals to access CPU with this pins
- Has 26 pin expansion header, 17 usable as GPIO
- Can be configured to support SPI, PWM, I2C buses



- Chip Select (/CS) Enables and disables device operation
- Serial Data Input (DO, DI) serially write instructions, address or data to the device on the rising edge of Serial Clock
- Serial Clock (SCK) provides timing for I/O operations
- Ground and Power 3.3V




| RPi header | SPI flash |
|---|---|
| 17 | Vcc 3.3V |
| 25 | GND |
| 19 | DI |
| 21 | DO |
| 23 | SCK |
| 24 | /CS |

# Basic Procedure

- Connect serial console cable from GPIO pins to USB port on laptop
- Connect each IC hook to GPIO pin and clip to SPI chip
- Power on
- Run minicom
- Run flashrom

```
Command line (5 args): flashrom -r intel-ct-factory-flash.bin -V -p
linux_spi:dev=/dev/spidev0.0
Calibrating delay loop... OS timer resolution is 5 usecs, 225M loops per second, 10 myus =
11 us, 100 myus = 116 us,                                    1000 myus = 1009 us, 10000
myus = 10053 us, 20 myus = 21 us, OK.
Initializing linux_spi programmer
Using device /dev/spidev0.0
The following protocols are supported: SPI.
Probing for AMIC A25L05PT, 64 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for AMIC A25L05PU, 64 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
.
.
.
probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for SST unknown SST SPI chip, 0 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for ST unknown ST SPI chip, 0 kBprob_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for Sanyo unknown Sanyo SPI chip, 0 kB: probe_spi_rdid_generic: id1 0xef, id2
0x3013
Probing for Generic unknown SPI chip (RDID), 0 kB: probe_spi_rdid_generic: id1 0xef, id2
0x3013
Probing for Generic unknown SPI chip (REMS), 0 kB: probe_spi_rems: id1 0xef, id2 0x12
Found WinboX" (512 kB, SPI).
Reading flash... done.

real    0m11.874s
user    0m0.690s
sys     0m0.150s
```

```
Command line (5 args): flashrom -w 808610d3.rom -V -p linux_spi:dev=/dev/spidev0.0
Calibrating delay loop... OS timer resolution is 5 usecs, 226M loops per second, 10 myus = 11 us,
100 myus = 109 us, 1000 myus = 1160 us, 10000 myus = 10299 us, 20 myus = 32 us, OK.
Initializing linux_spi programmer
Using device /dev/spidev0.0
The following protocols are supported: SPI.
Probing for AMIC A25L05PT, 64 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for AMIC A25L05PU, 64 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
.
.
.
id1 0xef, id2 0x3013
Probing for Sanyo unknown Sanyo SPI chip, 0 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for Generic unknown SPI chip (RDID), 0 kB: probe_spi_rdid_generic: id1 0xef, id2 0x3013
Probing for Generic unknown SPI chip (REMS), 0 kB: probe_spi_rems: id1 0xef, id2 0x12
Found Winbond flash chip "W25X40" (512 kB, SPI).
Reading old flash chip contents... done.
Erasing and writing flash chip... Trying erase function 0... 0x000000-0x000fff:EW, 0x001000-
0x001fff:EW, 0x002000-0x002fff:EW,
.
.
0x075000-0x075fff:S, 0x076000-0x076fff:S, 0x077000-0x077fff:S, 0x078000-0x078fff:S, 0x079000-
0x079fff:S, 0x07a000-0x07afff:S, 0x07b000-0x07bfff:S, 0x07c000-0x07cfff:S, 0x07d000-0x07dfff:S,
0x07e000-0x07efff:S, 0x07f000-0x07ffff:S
Erase/write done.
Verifying flash... VERIFIED.

real        0m27.106s
user        0m2.610s
sys         0m0.460s

making the flash image (as root):
git clone git://git.ipxe.org/ipxe.git
cd src
make bin/808610d3.rom
./util/padimg.pl --blksize=524288 --byte=0xff bin/808610d3.rom
```

# Questions??