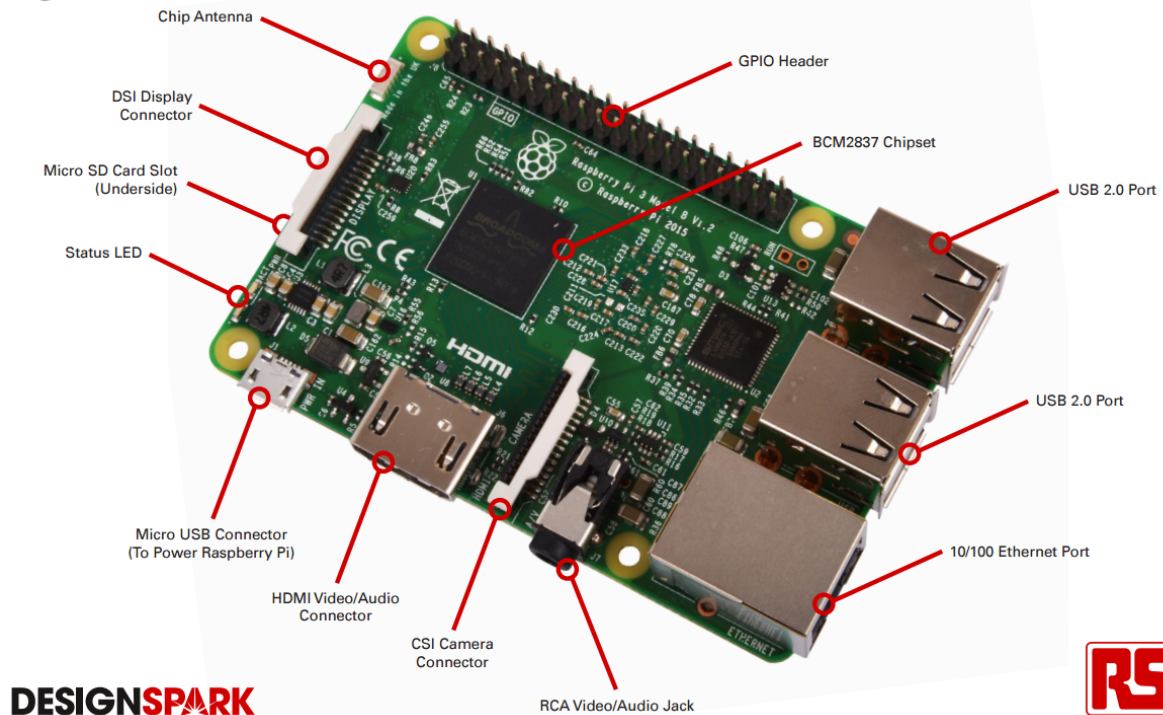




Raspberry Pi 3 Model B



DESIGNSPARK



- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- VideoCore IV 3D graphics core

OS: Raspbian Jessie med Pixel.

Namn: piOne, piTwo, piThree, piFour

Login

ssh pi@<ip>

raspberry

Elektronik

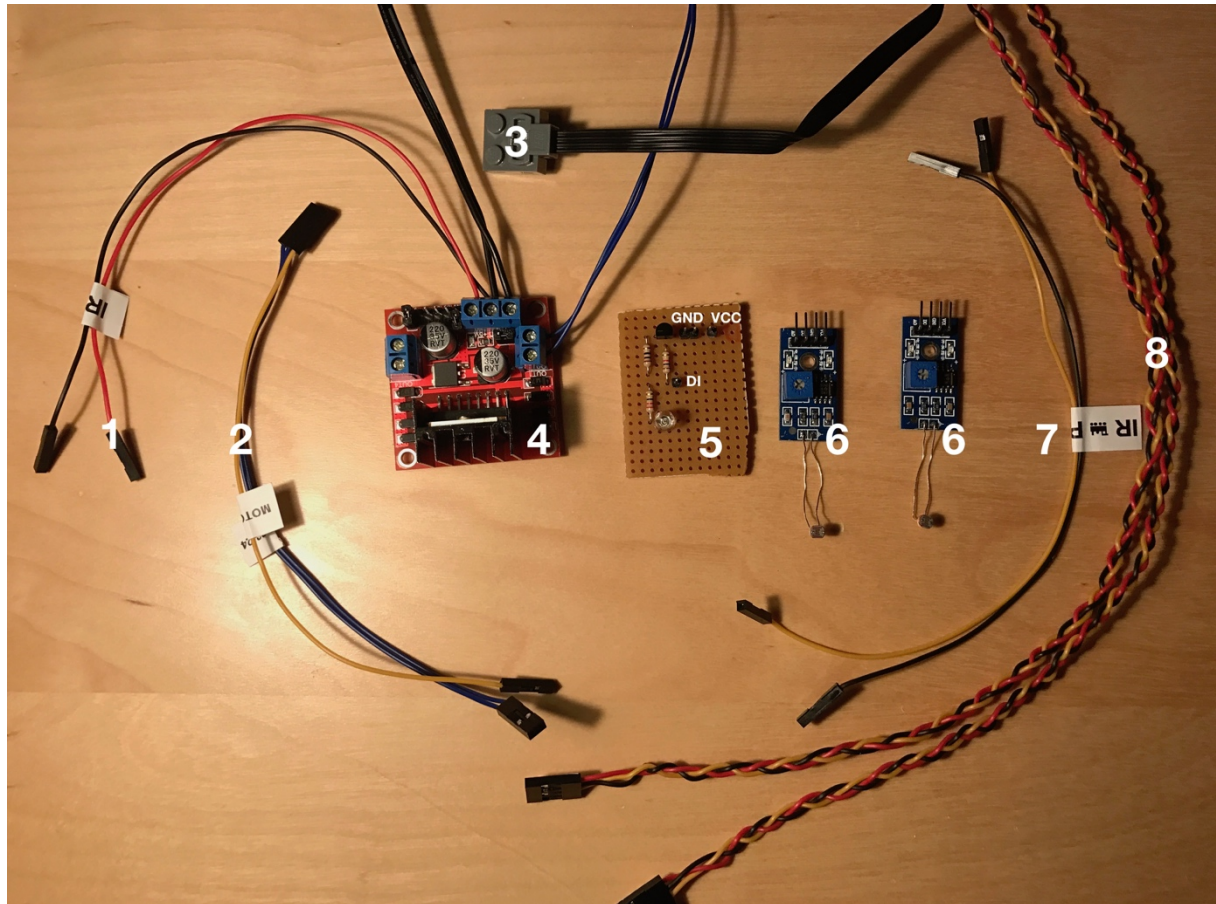
"Friskt kopplat, hälften brunnet"

Svart = Jord (GND)

Röd = 5V (VCC)

Gul = Signal - GPIO / DI / DO

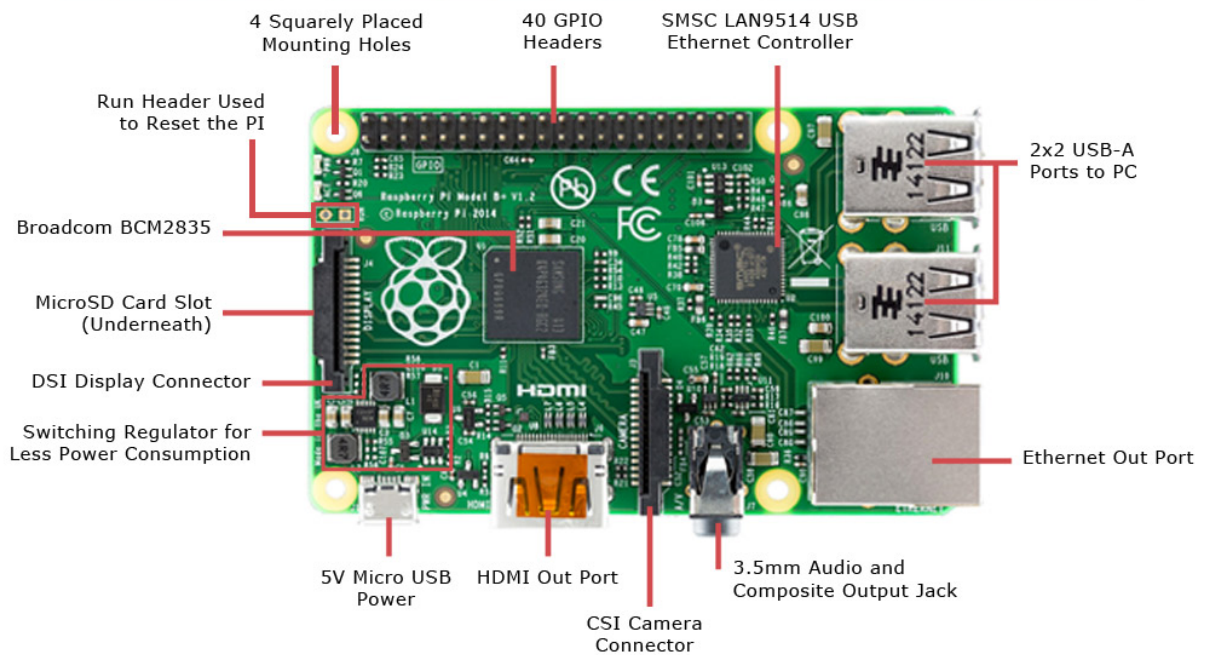
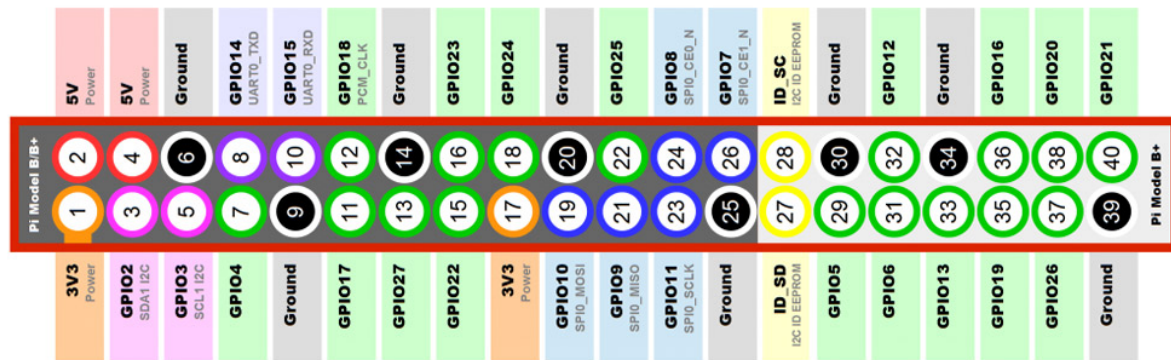
Blå = Signal - GPIO / DI / DO



1. 5V och jord till IR-fjärr (5)
2. Motorkabel. Stora kontakten till motorkontroller (4). Gul (GPIO 25), Blå (GPIO 23, 24) i Raspberry
3. Anslutning till motor / växel
4. Motorkontroller. Styr växelmotorn och ger 5V till IR-fjärr (5)
5. IR-fjärr
6. Ljussensor. Signalerar när tåget passerar
7. GND och Signal (GPIO 22 -> DI). Raspberry till IR-fjärr (5)
8. Kabel till ljussensor (6). GPIO 20 -> Sensor 1. GPIO 21 -> Sensor 2

GPIO Pinout Diagram

GPIO Pinout Diagram



Förslag på saker att göra

RASPBERRY PI WITH LINUX LESSONS

<http://www.toptechboy.com/raspberry-pi-with-linux-lessons/>

Labbar med tågen

Lab 1

Ansluta IR-diod för att styra tåg

Mål: Tåget ska gå att styra från terminalen

Lab 2

Styra tåget och få det att stanna när det passerar en ljussensor

Mål: Läs av sensor

Lab 3

Styra tåget och få det att stanna när det passerar ljussensor 1. När tåget passerar ljussensor 2, så ska växeln slå om.

Mål: Styra IR-fjärr och växel

Lab 4

Installera Node.js, bygga tjänster för att styra tåget

Mål: Tåget ska gå att styra via gui.

Lab 5

Fler sensorer. Ska ge återkoppling om position. Websockettjänster.

Mål: Bygga fler tjänster som skickar tillbaka realtidsdata.

Lab 6

Manöverpanel med touchskärm. En skärm som används för att styra och en för att visa realtidsinformation.

Lab 7

Styra två tåg på samma bana. Undvika kollisioner

Mål: Köra två tåg som växlas mellan spåren.

IR-fjärr kommandon

IR-fjärren styra med hjälp av LIRC.

```
irsend
```

Listar alla kommandon

```
irsend LIST LEGO_Single_Output ""
```

Kör tåget framåt

```
irsend SEND_ONCE LEGO_Single_Output 1B_4
```

```
irsend SEND_ONCE LEGO_Single_Output 1B_BRAKE
```

Växelkod – Python

```
import sys
import time
import RPi.GPIO as GPIO

StepPinForward=23
StepPinReverse=24
PowerPin=25
MotorPower=60

sleeptime=1

GPIO.setmode(GPIO.BCM)

GPIO.setup(StepPinForward,GPIO.OUT)
GPIO.setup(StepPinReverse,GPIO.OUT)
GPIO.setup(PowerPin,GPIO.OUT)

GPIO.output(StepPinForward, GPIO.LOW)
GPIO.output(StepPinReverse, GPIO.LOW)

my_pwm=GPIO.PWM(PowerPin,100)

def forward(x):
    GPIO.output(StepPinForward, GPIO.HIGH)
    my_pwm.start(MotorPower)
    print "forwarding running motor "
    time.sleep(x)
    GPIO.output(StepPinForward, GPIO.LOW)
    my_pwm.stop()

def reverse(x):
    GPIO.output(StepPinReverse, GPIO.HIGH)
    my_pwm.start(MotorPower)
    print "backwarding running motor"
    time.sleep(x)
    GPIO.output(StepPinReverse, GPIO.LOW)
    my_pwm.stop()

forward(0.5)

print "Stopping motor"
GPIO.cleanup()
```

Sensorkod – Python

```
from time import sleep      # Import time library
import RPi.GPIO as GPIO     # Import RPi Library
GPIO.setmode(GPIO.BCM)     # Specify We Want to reference
Physical pins

sensor1=20  # Descriptive Variable for pin 20
sensor2=21  # Descriptive Variable for pin 21

GPIO.setup(sensor1,GPIO.IN) # Set sensor1 as input and
Activate pull up resistor
GPIO.setup(sensor2,GPIO.IN) # Set sensor2 as input and
Activate pull up resistor

while(1): # Create an infinite loop

    if GPIO.input(sensor1)==1: # sensor 1 will report 0 if
it is passed
        print "Sensor 1 Passing"
        sleep(.1)  # delay
    if GPIO.input(sensor2)==1: # sensor 2 will report 0
if it is passed
        sleep(.1)
        print "Sensor 2 Passing"

GPIO.cleanup()
```

Referenser

Grunden till tågidéen

<http://www.internetoflego.com/lego-train-automation-ir-power-functions-with-nodejs-and-lirc/>

Beskrivning av pinnarna på RaspberryPi

<https://pinout.xyz/>

RASPBERRY PI WITH LINUX LESSONS

<http://www.toptechboy.com/raspberry-pi-with-linux-lessons/>

Lego-LIRC

<https://github.com/dspinellis/lego-lirc>

Controll LIRC from NodeJS

https://github.com/alexbain/lirc_node

https://github.com/alexbain/lirc_web

<https://github.com/pmgration/node-infrared>