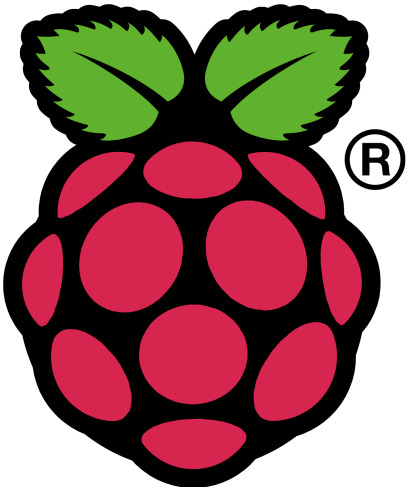
# Controlling A Remote Control With The Piface Digital

Document Prepared By
A'mmer Almadani
[Mad_Dev]



*Raspeberry Pi Logo*[2]

*PiFace Logo*[1]

1 http://www.piface.org.uk
2 http://www.raspberrypi.org

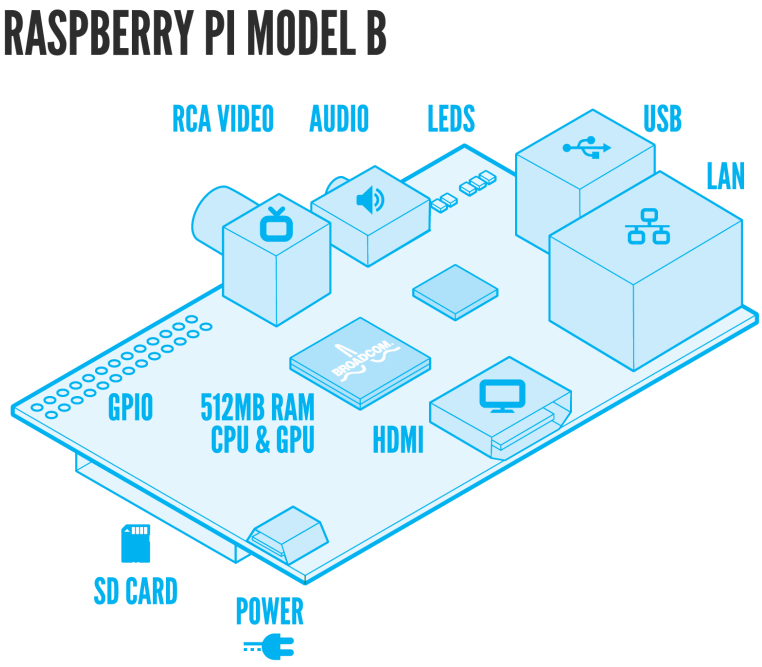
## Raspberry Pi:

The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.

The Raspberry Pi is manufactured through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Egoman. These companies sell the Raspberry Pi online. Egoman produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pis by their red coloring and lack of FCC/CE marks. The hardware is the same across all manufacturers.

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC),which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and persistent storage. The Foundation's goal was to offer two versions, priced at US$25 and US$35. They started accepting orders for the higher priced model B on 29 February 2012, and the lower cost model A on 4 February 2013.

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, and Perl.[3]




*Raspberry Pi Model B*

3 https://en.wikipedia.org/wiki/Raspberry_Pi

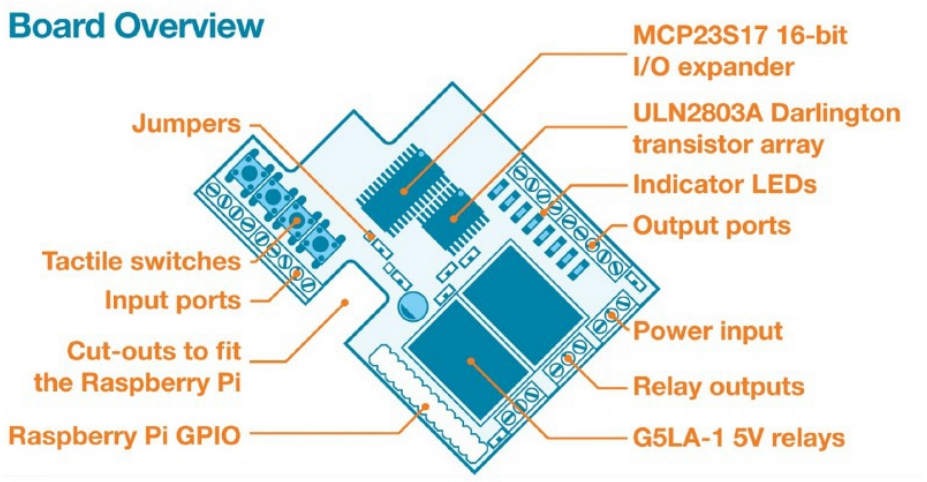
## Piface Digital:

PiFace Digital is designed to plug on to the GPIO of your Raspberry Pi, allowing you to sense and control the real world.

With PiFace Digital you can detect the state of a switch, for example from a door sensor, a pressure pad or any number of other switch types. Once this state has been detected, you can write your own software for the Raspberry Pi that determines how to respond to that switch state. You can drive outputs to power motors, actuators, LEDs or anything you can imagine.

### Features:

- Plugs directly onto the Raspberry Pi GPIO socket
- Fits Directly over the Raspberry Pi and within the Raspberry Pi's footprint
- 2 Changeover Relays
- 4 Tactile Switches
- 8 Digital Inputs
- 8 Open-Collector OutPuts
- 8 LED Indicators
- Easy to program in Python, Scratch and C
- Graphical Emulator and Simulator
- Relays can be used to switch voltages up to 20V (Max) or currents up to 5A (Max) [4]




*Piface Digital*

4 http://www.piface.org.uk/products/piface_digital/

## Introduction:

There is no denying that the PiFace Digital[5] is an essential peripheral to the Raspberry Pi, especially to those inept with electronic circuitry. Before the PFD was introduced, I used to request help from friends whose adequacy with electronics surpassed mine; it proved annoying that I would have to wait for a friend to instruct me on the next step of the build. But with PFD's deployment, my only focus was on software; abandoning the hindrance of schematics and wiring. For that, I sincerely thank the team behind the PFD.

This tutorial will hopefully guide you through the process of 'hacking' any[6] remote control and eventually, virtually controlling it at your will.

The remote used in this example controls an air conditioning unit; a Mitsubishi MS-18JN-S1[7]. It is an old unit; I doubt that you will have the exact remote. However, the principle stands, across all remote controls.

## Addendum:

- The raw version of the Test-Script can be found at
http://emoticode.net/python/rpi-piface-controlling-an-ac-units-remote-control.html

- In case you do not have the same remote, trial&error.

- This tutorial is not intended for practical use, rather a showcase of the PFD.

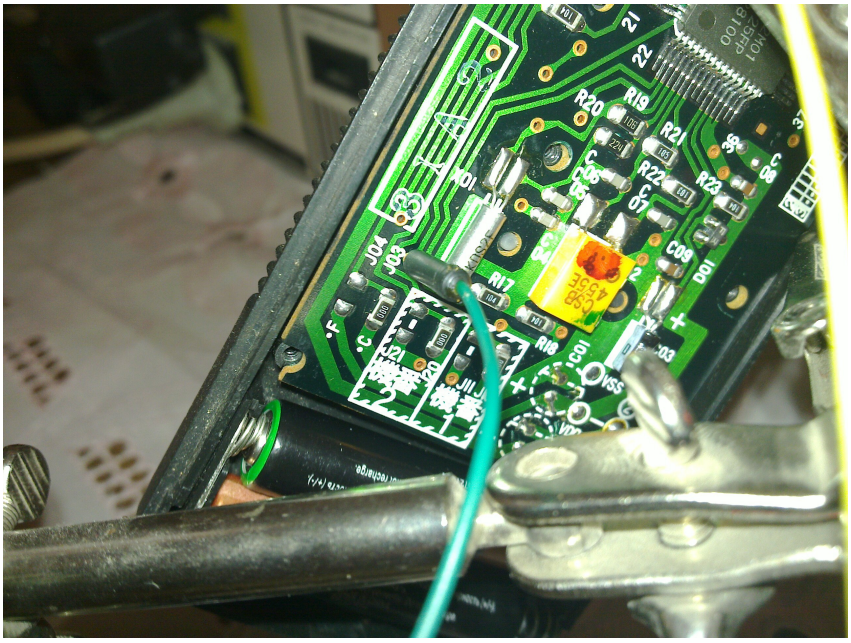

*MS-18JN-S1 Remote Control*[8]

5 Hereon known as PFD
6 Given enough time, any remote could be hacked
7 http://www.mylinkdrive.com/uploads/documents/4309/document/OB188_-_MS18-24NN_Technical.PDF
8 Please forgive its dire state

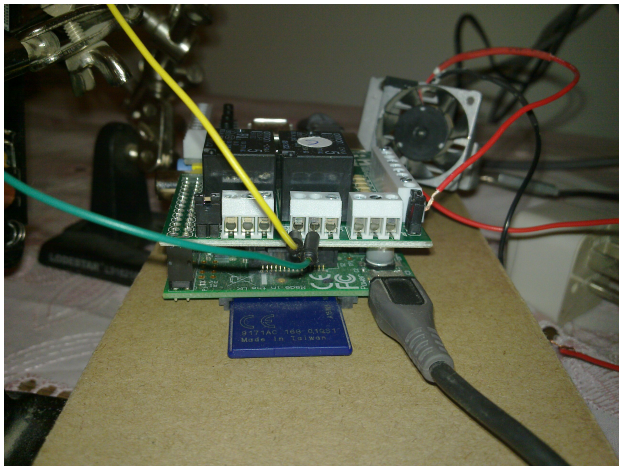
## Tutorial:

Objective: *Start/Stop the AC unit using the Raspberry Pi*
Difficulty: *easy*
Duration: *30 minutes*

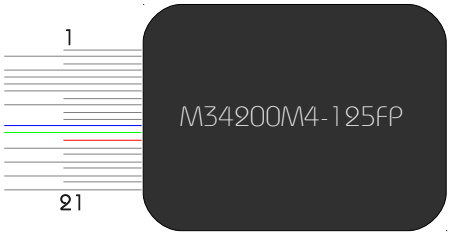
- Unscrew The remote
- Locate the power button – in this example, the power button has multiple connections, J03 was the closest one(trial&error).
    - It will require you trial&error tests to fully comprehend how the system operates.
    - If the remote you wish to manipulate is not as sophisticated, using a multimeter/ohm-meter(continuity test), try to find the two locations that will close the circuit(usually the common -/+ and the button you wish to emulate)

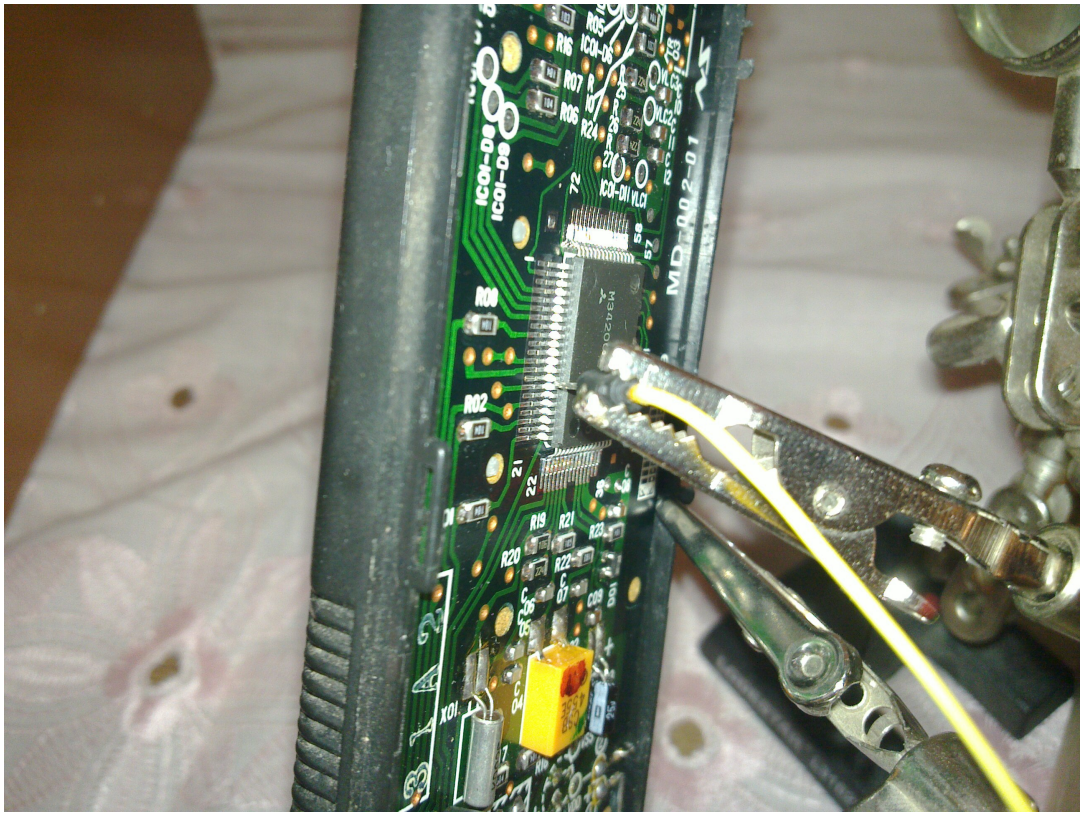



- Connect a wire to J03(Green-wire)
- Connect the other end to the relay output pin in the PFD(Green-wire)



- Connect a wire to Pin 14 on the remote (IC) (Yellow-wire)
    - During my trail&error phase, I discovered the following; retaining J03 is connected :
        - Pin 12 – Temperature select (Blue)
        - Pin 13 – Mode switch (Green)
        - Pin 14 – On/Off (Red)



- Connect the other end to the relay output pin in the PFD(Yellow-wire)
- Start the Raspberry Pi
    - Test:

```
pi@raspberrypi$ python 3
>> import pifacedigitalio as pf
>> pfd = pf.PiFaceDigital()
>> pfd.relays[1].value = 1
```

        - The monitor on the remote should turn on(if turned off) or turn off(if turned on)

## Test-Script:

```
#!/usr/bin/env python3

import pifacedigitalio as pf
import time
import subprocess
import urllib
from xml.dom import minidom

def get_temp(cc):
	##Thanks to http://stackoverflow.com/questions/16070885/python-yahoo-weather-xml-parse-works-
with-2-7-1-but-not-2-6-1
	CITY_ID = cc
	TEMP_TYPE = 'c'
	WEATHER_URL = 'http://xml.weather.yahoo.com/forecastrss?w=' + CITY_ID +' &u=' + TEMP_TYPE
	WEATHER_NS = 'http://xml.weather.yahoo.com/ns/rss/1.0'
	dom = minidom.parse(urllib.urlopen(WEATHER_URL))
	ycondition = dom.getElementsByTagNameNS(WEATHER_NS, 'condition')[0]
	CURRENT_OUTDOOR_TEMP = ycondition.getAttribute('temp')
	return CURRENT_OUTDOOR_TEMP
def switch_control():
	'''You can think of this a virtual toggle switch.
	   The value for ON and OFF is 1(one)
	   I am not sure why, but I am assuming it is due to the method of wiring(IC pin)
	   The initial state is imperative for the operation. e.g. If the remote is on, the switch
will close it.
	'''
	pfd = pf.PiFaceDigital()
	pfd.relays[1].value = 0
	time.sleep(0.7) #I read somewhere that it is not recommended to switch the relay immediately
	pfd.relays[1].value = 1
def run():
	cc = '2346951'
	start = '18'
	stop = '17'
	state = remember()
	print('The Current State is '+state)
	print('Based on Yahoo! Outdoor Temp = %s C\n' %get_temp(cc))
	if get_temp(cc) >= start:
		print('Temp is at start range [%s]' %start)
		if 'on' in state:
			print('AC is on') #Not the actual AC, rather the remote
			pass
		elif 'off' in state:
			print('Starting...')
			switch_control()
			b = open('state', 'w+')
			b.write('on')
			b.close()
	if get_temp(cc) <= stop:
		print('Temp is at stop range [%s]' %stop)
		if 'off' in state:
			print('AC is off')
			pass
		elif 'on' in state:
			print('Stopping')
			switch_control()
			b = open('state', 'w+')
			  b.write('off')
			  b.close()
	else:
		pass
def remember():
	'''You need to enter the first state manually
	   1- on
	   2- off
	'''
	a = open('state', 'r')
	state = a.read()
	a.close()
	return state
run()
```