

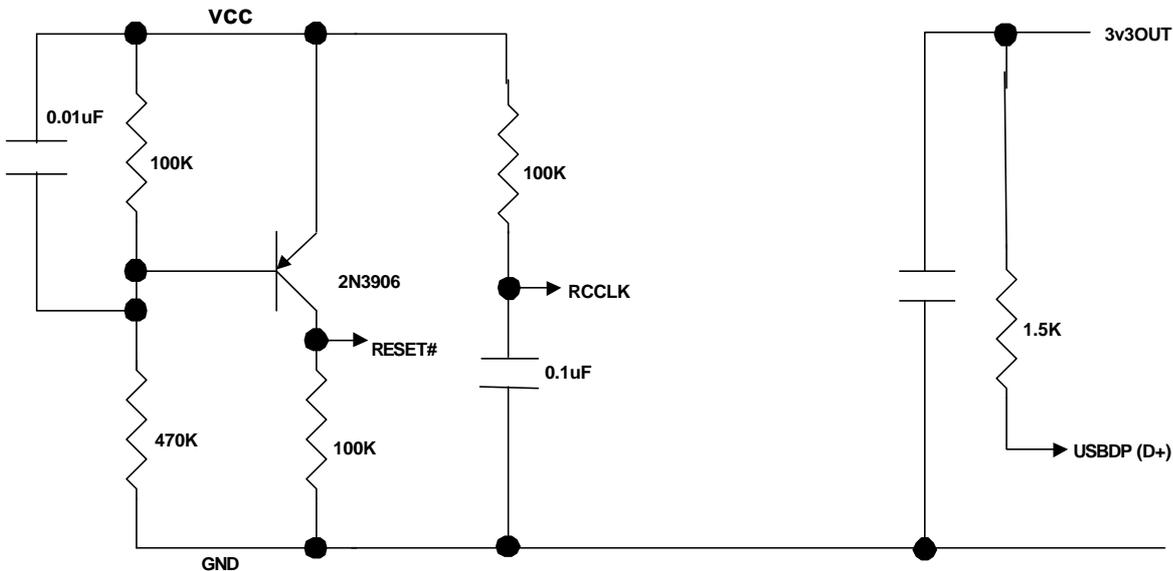
## Bus Powered / Self Powered Interface Circuits

*This document gives a few examples of ways to connect the FT8U232 or FT8U245 chip for different power and reset schemes.*

### BUS POWERED OPTIONS

#### 1 . The standard bus powered schematic

Connections for RESET#, RCCLK and USBDP in bus powered mode



In this circuit, the reset is driven by a transistor which is biased so that the device is held in reset until VCC > 3.6 volts. This ensures a reliable reset. A 10nF capacitor provides an additional RC time delay of around 2ms.

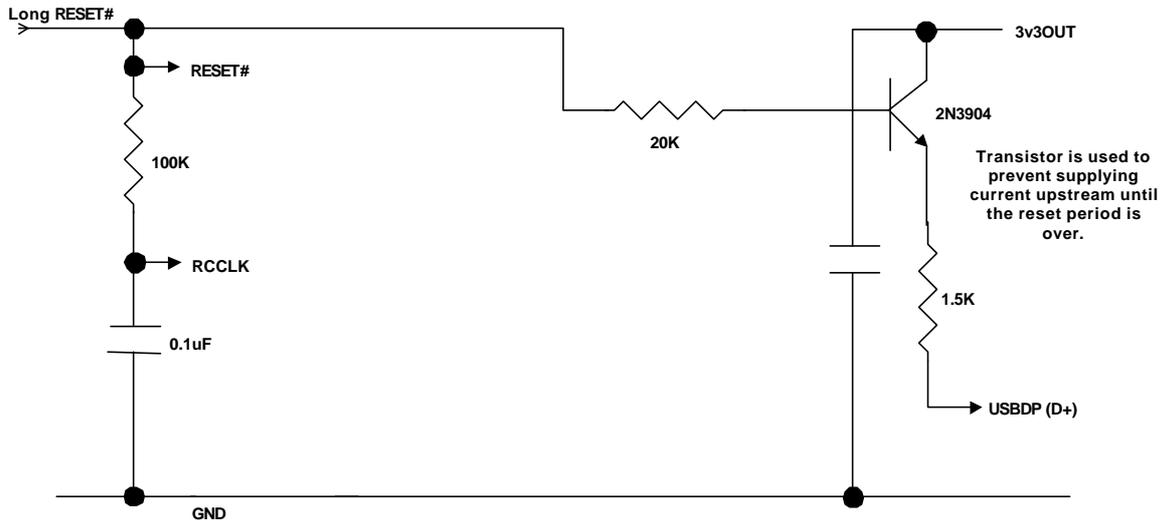
This circuit will work for both 6 or 48 MHz operation.

Important points to note about the standard configuration –

- A. The reset time is under 10mS so no special care is needed regarding USB trying to enumerate the device whilst it is still in reset ( see below )
- B. The component values are chosen such that RESET# will go inactive before RCCLK ( see below )

### 2 . Bus powered schematic with extended reset period

Connections for RESET#, RCCLK and USBDP in an extended RESET# bus powered mode



If using a dedicated reset i.c. ( MAX809 etc ) many of whom have a extended reset time of hundreds of milliseconds or when resetting the device from a MCU or other logic, there are two areas which require careful consideration –

1. The 1.5k resistor on D+ indicates to USB that a full speed device is connected when the device is hot plugged into a USB hub or host port. If the reset time is > 10ms from plugging in the device, then there is a possibility that the device will still be held in reset at the time that USB tries to enumerate the device. This will result in USB marking the device as an “unknown device” as it is unable to enumerate it. The solution in this case is to connect a NPN transistor in series with the 1.5k pull up so that the USB D+ line is not pulled high until after the device comes out of reset. The base of the transistor is connected to RESET# via a current limiting resistor.

2. For reliable 6MHz operation RCCLK must only go high after RESET# goes inactive. Failure to do this will result in the internal x8 clock multiplier not always being enabled correctly. The fix is simple – connect the 100k pullup on the RCCLK circuit to RESET# instead of VCC. That way, RESET# will always go high before RCCLK regardless of the period of the RESET time.

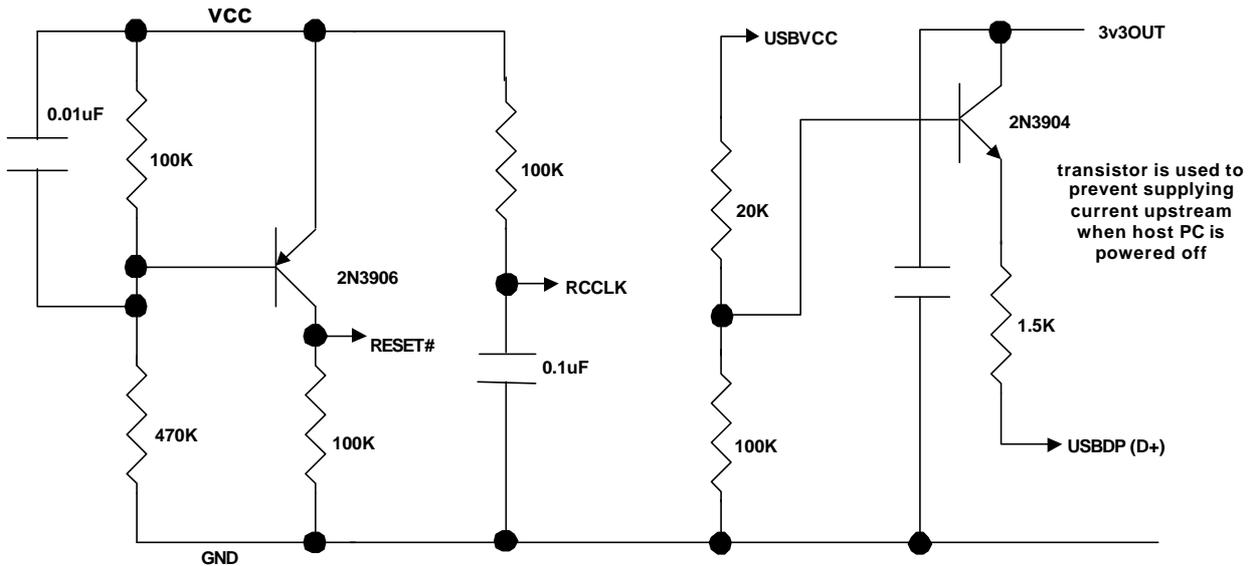
#### Important Note on using external Reset I.C.'s

When using a reset i.c. in this configuration select one that can drive out both high and low. Some of these devices have open-collector like outputs that can only pull low.

## SELF POWERED OPTIONS

### 1. The Standard Self Powered Schematic

Connections for RESET#, RCCLK and USBDP in a self powered mode



In this case the RESET# and RCCLK circuits are the same as for the standard bus powered version.

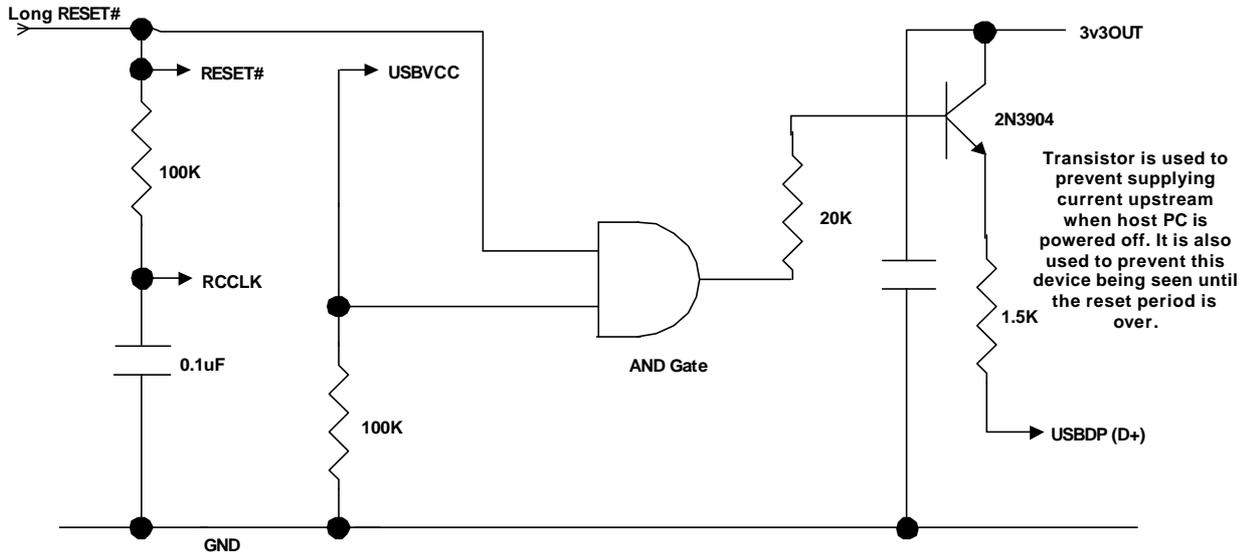
The USB specification states that a peripheral device is not allowed to source power into a USB host or hub port when the PC or hub is powered down. When a device is self powered ( has it's own PSU ) it would be possible for the device to force current down the D+ line via the 1.5k resistor on USB D+ when the PC ( or hub ) is turned off and the device is still powered up through it's own PSU. Failure to pay attention to this can cause unreliable enumeration on some hosts and hubs.

In the above diagram, VCC is supplied from the self-powered device's PSU whilst USBVCC refers to the voltage supplied by VCC on the power pin of the upstream connector. The solution is to connect an NPN transistor in series with the 1.5k pull up resistor on USB D+, getting it's bias current from USBVCC as shown above. The voltage from the upstream connector will turn the transistor off and on. If it is unplugged or the host PC is off, the transistor is turned off and then there is no current supplied to the D+ line through the 1.5k pull up.

## High Speed USB Controllers for serial and FIFO applications

### 2. The Self Powered Schematic with extended RESET# period.

Connections for RESET#, RCCLK and USBDP in an extended RESET# bus powered mode



Please refer to the “Bus powered schematic with extended reset period” notes ( above ) for an explanation of the issues relating extending the reset period.

In this example, RESET# is coming from a source, where it is longer than 10 milliseconds such as a reset generator i.c or a MPU IO pin etc. As a result, the USB D+ line cannot be pulled high immediately on powering up the host, but has to wait until the extended reset period is over. Again, we use the familiar series transistor circuit on USB D+ but this time the transistor has to be turned off under two conditions –

1. USB Power from the host or hub is off ( see above for why )
2. The device is still being held in reset.

The solution outlined here uses a CMOS AND gate to achieve the desired function, though it is also possible to achieve the same function using all discrete components also. Also note that the 100k pull-up on RCCLK is connected to RESET# and not to VCC. The reason for this has been explained in the Bus Powered section.