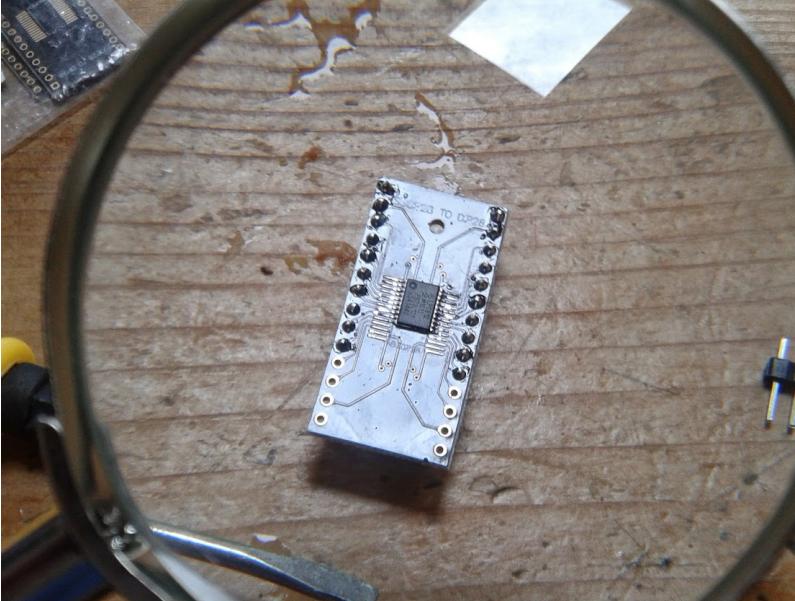


Baremetal programming of the LPC812 from NXP

Introduction

At the time this page was written (September 2014), the LPC1114FN28 processor was hard to get in Europe. The only breadboard friendly ARM Cortex microcontroller that you can purchase locally is the LPC810. This chip is useful enough but suffers from a lack of pins. The LPC812M101JDH20FP uses the same core as the LPC810, has more memory, pins and peripherals however it is in a TSSOP-20 package - not much use from a breadboarding perspective. Various suppliers in Europe and further away will ship you a TSSOP to DIP adapter. I got some in dx.com (about 1 Euro each). With some careful soldering (use plenty of flux) it is pretty easy to mount the TSSOP-20 LPC812 onto the adapter. I used a TSSOP to DIP-28 adapter (ignore the extra pins). The pictures below shows my efforts (right click to view image and zoom-in).



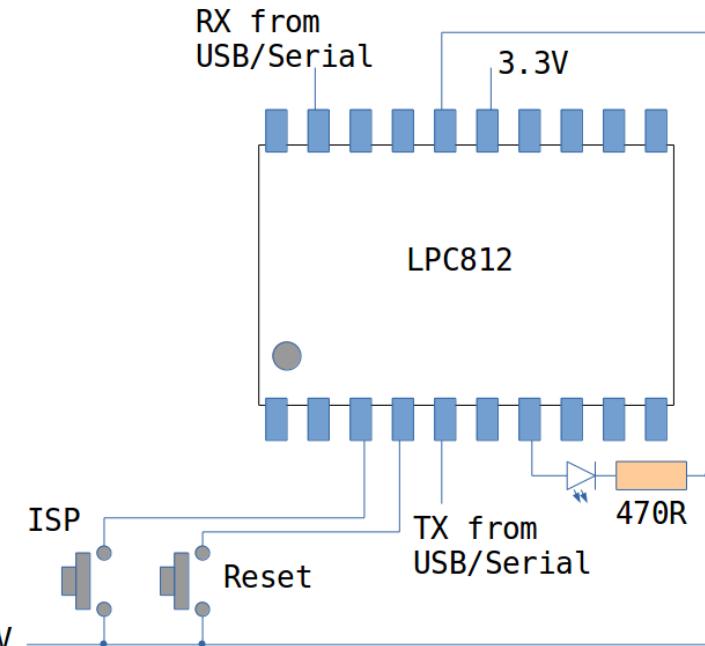
Soldering complete, it was now possible to mount the 20 pin lpc812 on a breadboard.

Blinky

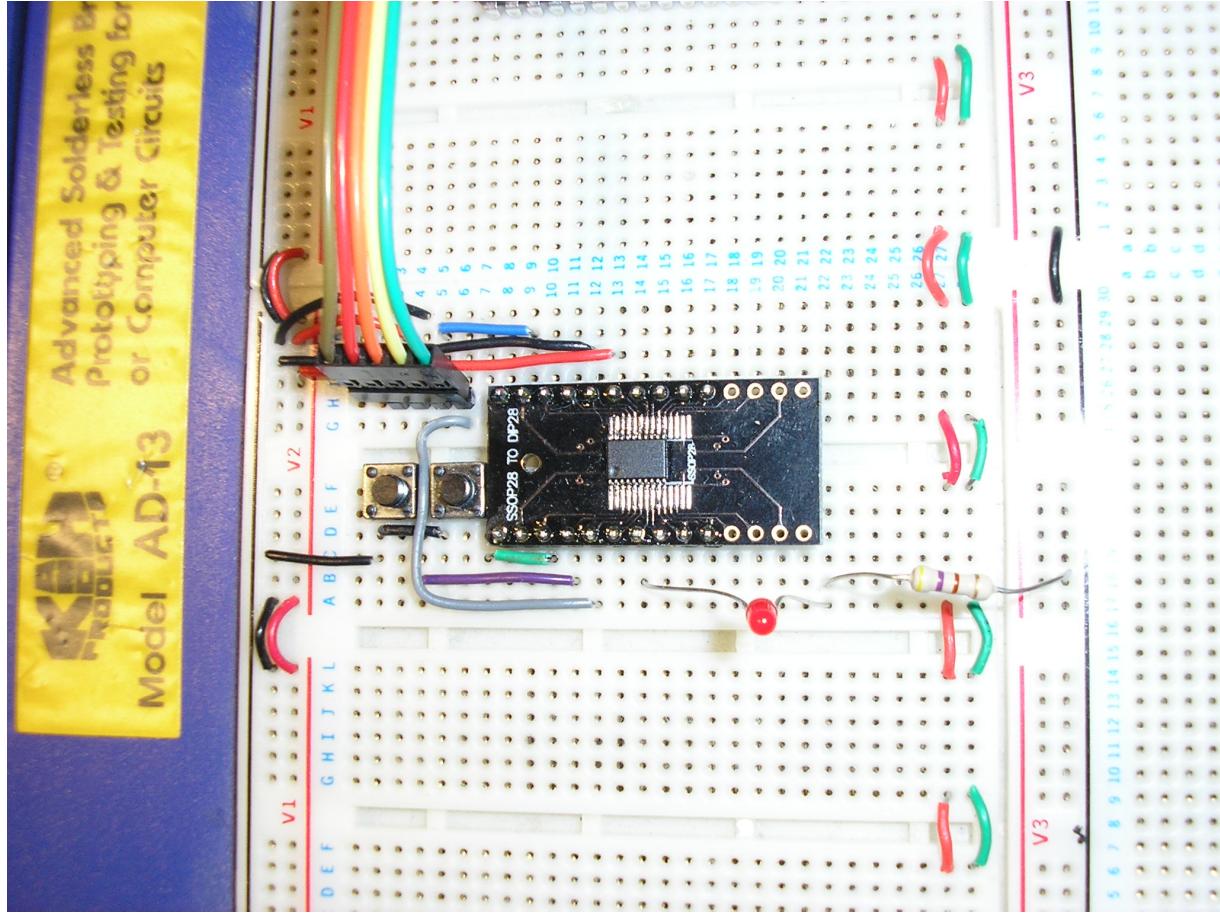
This program simply flashes an LED. The code can be compiled using arm-none-eabi-gcc (GCC for ARM devices without an OS) which can be obtained [here](#).

All support files are included in the code so a complex IDE/Toolchain is unnecessary. Header files and linker scripts were written using information in the LPC81x user manual (document UM10601) and the LPC81xM datasheet. The device is programmed using the lpc2isp utility (<http://lpc2isp.sourceforge.net/>) and a USB to serial converter (3.3 volts). TX from the USB/Serial device goes to pin 5 on the LPC812; RX from the USB device goes to pin 19 (Refer to table 6 in the LPC81xM datasheet). The chip is also powered from the USB/Serial converter's 3.3V and 0V outputs. To program the device you must pull its "Reset" (pin 4) and "ISP Entry Pin" (pin 3) low. Release Reset first followed by the ISP entry pin. The device is now in ISP mode and you can download your program using lpc2isp.

The schematic of the circuit is as follows:



A picture showing the circuit is below



[Blinky source code and support files](#)

Other examples

[Serial port communications with support files](#). This example uses interrupt driven reception and polled transmission
(more to follow)

[Back to my ARM project page](#)