User firmware in the Twinklers

Version 1.0
November 2009

This application note describes the electronical configuration of the "Twinkler" hardware, and how to store user firmware in it.

Contents

ITB CompuPhase

Trademarks

Copyright

The examples and programs in this application note have been included for their instructional value. They have been tested with care, but are not guaranteed for any particular purpose.

Introduction

A Twinkler consists of a low-power RGB LED and a LED controller on a small "printed circuit board" (PCB). The controller handles the serial protocol, by which the colour of the LED and fading & blinking settings are transmitted from a PC to each Twinkler. The controller also handles the refreshing of the LED, via "pulse width modulation" (PWM). A separate document covers the standard operation of the Twinklers, and interfacing a chain of Twinklers to a PC, in detail.

The controller is a micro-controller with its program stored in Flash memory —called firmware. Although the Twinkler is flexible in its standard configuration, you may think of features, extensions or alternate uses of the Twinkler hardware that are beyond the capabilities of the embedded firmware. If this is the case, you can create and store new firmware into the Twinkler's micro-controller. This application gives the required information to do so.

Writing new firmware requires an assembler or a compiler for the micro-controller on the Twinkler. The section "Twinkler hardware" gives the details about this part. Storing the new firmware on a Twinkler furthermore requires a device called an "in-circuit programmer" and an adaptor cable. Various kinds of in-circuit programmers exist on the market, but you must construct the adaptor cable yourself. The section "In-circuit programming" lists a few acceptable programmers and covers the construction of the cable.

Caution: Before getting into details of how to write and store new firmware in the Twinklers, please observe that ITB CompuPhase offers **no support** for non-standard firmware, and **no warranty** for Twinklers with modified firmware. Faulty firmware **may damage** the controller or other logic on the Twinkler board.

Twinkler hardware

The Twinkler has a PIC micro-controller from Microchip, specifically the PIC16F688. For purpose of programming this micro-controller, it is assumed that you have obtained the relevant technical documentation from Microchip's web site.

The PIC16F688 micro-controller runs at 20 MHz, using an on-board ceramic resonator between pins 2 and 3. Pins 5 and 6 are used for the serial connection (Rx and Tx respectively). Pins 8, 9 and 10 are connected to the green, blue and red channels of the RGB LED. Pins 4, 12 and 13 are used for ICSP programming. Pins 7 and 11 are left unconnected. Finally, pins 1 and 14 are connected to power and ground respectively. There is a 47Ω resistor between pins 5 and 12, and another 47Ω resistor between pins 6 and 13.

In practice, it means that only the pins 4, 5/12 and 6/13 of the PIC16F688 are available for external inputs or outputs, because only these pins are represented on the two Micro-MaTch connectors ("INPUT" and "OUTPUT"). Pins 12 and 13 are wired to pins 5 and 6 respectively.

Below is a table with a summary of the connections between the CPU pins and the Micro-MaTch connectors, or fixed functionality for the CPU pins.

PIC16F688	Connected to	Notes
1	+5V	
2	Resonator	Ceramic resonator, 20 MHz
3	Resonator	idem

4	OUTPUT:3	internal (weak) pull-up resistor
5	INPUT:4	through 47Ω resistor
6	OUTPUT:4	through 47Ω resistor
7		not connected
8	LED	green channel
9	LED	blue channel
10	LED	red channel
11		not connected
12	INPUT:4	direct connection (no resistor)
13	OUTPUT:4	direct connection (no resistor)
14	ground	

In-circuit programming

The PIC micro-controller is programmable with the Microchip ICSP protocol, by hooking up the programmer to *both* Micro-MaTch connectors. A Twinkler must be removed from any daisy-chain before it can be programmed. Each Twinkler must be programmed individually, chain programming is not possible.

For programming, you will need a PIC programmer capable of "ICSP" (in-circuit serial programming). Of course, the programmer also needs to support the particular micro-controller: the PIC16F688. We have successfullyn used the following product:
◇ **PICkit 2** programmer/debugger by Microchip; a low-cost device targetted at developpers.
◇ **Load-n-Go** handheld programmer by CCS, a battery-powered portable programmer.

PIC programmers that support ICSP have a six-pin connector. These pins must be connected to several pins of the two Micro-MaTch connectors of the Twinkler. Although all ICSP-capable PIC programmers can provide power to a board, it is advised to power the Twinkler with an external power supply —the PIC programmer supplies approximately 5V on its connector, but Twinklers should be powered with 5.5V minimum.

The table below gives the suggested cable lay-out between the ICSP connector and the Micro-MaTch connectors of the Twinkler.

ICSP	**Twinkler**	**Notes**
1	OUTPUT:3	
2		not connected
3	INPUT:6	
4	OUTPUT:4	
5	INPUT:4	
6		not connected
Battery		
+	INPUT:1	e.g. +9V DC
-	INPUT:6	also connected to ICSP pin 3

If you have a PIC programmer with memory for firmware (the "programmer-to-go" option), you may wish to make a slightly more elaborate circuit, including a power regulator to power the PIC programmer from the same 9V battery as the Twinkler's.

Index

- Names of persons or companies (not products) are in *italics*.
- Function names, constants and compiler reserved words are in `typewriter font`.