Attiny HV and ISP Programmer

Developed at TAMI

Tel-Aviv Makers International

Project goals

- Main goal
  - High Voltage fuse reset of ATTami boards.
- By products:
  - High Voltage fuse reset of Attiny components.
  - ISP programming of ATTami boards.
  - ISP programming of Attiny components.
  - ISP programming of various AVR based boards, under either 5V or 3.3V supply.
  - Acceptance test of ATTami boards.
  - Self contained ATTami bootloader loading (Not yet implemented)

Project structure

- Arduino shield with the following features:
  - Charge pump circuit to provide the 11.5V for the HV programming.
  - Bi directional 5V to 3.3V level shifters.
  - Sockets to enable programming of ATTami boards and Attiny components.
  - Indicator LEDs.
  - ATTami ATP LEDs.
  - Control switches.

Charge pump circuit

Charge pump is a circuit that produces high voltage by switching of capacitors.

See: http://en.wikipedia.org/wiki/Charge_pump

In our case we produce 11.5V out of the Arduino 5V supply.

The 11.5V is controlled and regulated by the Arduino.

Bi directional 5V to 3.3V level shifters

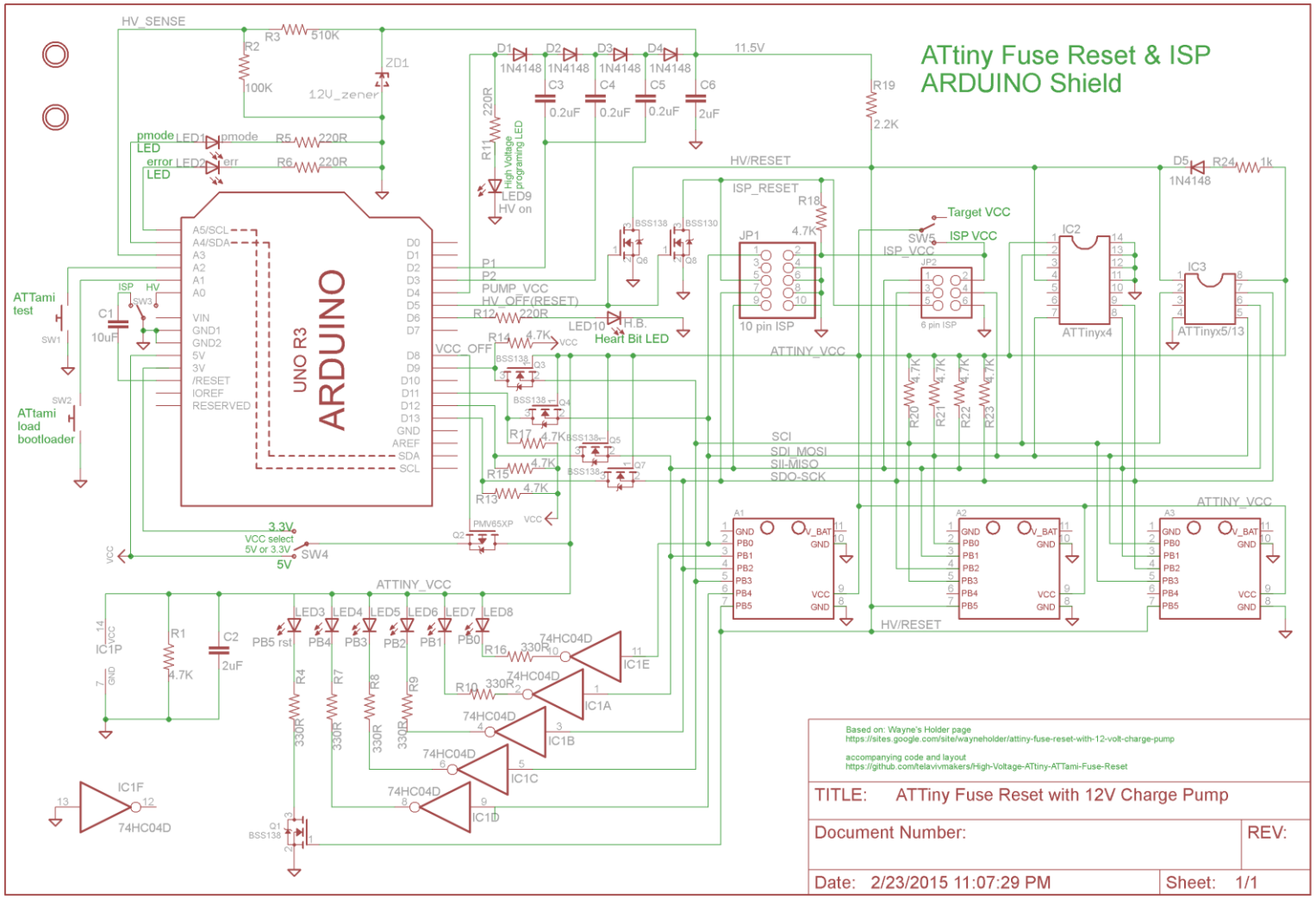
- 4 MOSFET based bi-directional level shifters to shift the 5V Arduino I/O pins level to 3.3V.
- See: http://delphys.net/d.holmes/hardware/levelshift.html

Sockets to enable programming of ATTami boards and Attiny components

- Pogo-Pins socket to enable programming and testing of pinless ATTami boards.
- Sockets to enable programming and testing of ATTami boards with upper or lower pins.
- Socket for 8 pin ATtiny programming.
- Socket for 14 pin ATtiny programming.
- Socket for 6 pin ISP ribbon.
- Socket for 10 pin ISP ribbon.

Control Switches

- ISP – HV selector: Selects witch programming mode will be performed
- 3.3V – 5V selector: Selects the I/O pins level to conform with the external board connected to the ISP cables.
- ISP Vcc – Target Vcc selector: Selects if The programmer will provide Vcc to the target board via the ISP cables.
- ATTami test pushbutton: Provides Vcc to the ATTami board.
- ATTami load bootloader: Initiate loading to the ATTami of the bootloader Hex file stored on the Arduino Flash memory. (This function is not yet implemented.)


ATTiny Fuse Reset & ISP
ARDUINO Shield
HV_SENSE
R3 510K
R2 100K
ZD1
12V_zener
D1 1N4148
D2 1N4148
D3 1N4148
D4 1N4148
11.5V
C3 0.2uF
C4 0.2uF
C5 0.2uF
C6 2uF
R19 2.2K
R11 220R
High Voltage programing LED
LED9 HV on
pmode LED
error LED
R5 220R
R6 220R
HV/RESET
ISP_RESET
R18 4.7K
D5 1N4148
R24 1k
Target VCC
ISP VCC
SW5
ISP_VCC
JP1
10 pin ISP
JP2
6 pin ISP
IC2
ATTinyx4
IC3
ATTinyx5/13
UNO R3 ARDUINO
A5/SCL
A4/SDA
VIN
GND1
GND2
5V
3V
/RESET
IOREF
RESERVED
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
GND
AREF
SDA
SCL
P1
P2
PUMP_VCC
HV_OFF(RESET)
R12 220R
LED10 H.B. Heart Bit LED
VCC_OFF
R14 4.7K
R17 4.7K
R15 4.7K
R13 4.7K
ATTiny test
ATTami load bootloader
ISP
HV
SW3
SW1
SW2
C1 10uF
3.3V
VCC select 5V or 3.3V
SW4
5V
ATTINY_VCC
SCI
SDI_MOSI
SDI-MISO
SDO-SCK
R20 4.7K
R21 4.7K
R22 4.7K
R23 4.7K
LED3
LED4
LED5
LED6
LED7
LED8
PB5 rst
PB4
PB3
PB2
PB1
PB0
R16 330R
R10 330R
R4 330R
R7 330R
R8 330R
R9 330R
74HC04D
IC1E
IC1A
IC1B
IC1C
IC1D
IC1F
IC1P
R1 4.7K
C2 2uF
Q1 BSS138
Q2 PMV65XP
Q3 BSS138
Q4 BSS138
Q5 BSS138
Q6 BSS138
Q7 BSS138
Q8 BSS130
A1
A2
A3
GND
PB0
PB1
PB2
PB3
PB4
PB5
V_BAT
VCC
Based on: Wayne's Holder page
https://sites.google.com/site/wayneholder/attiny-fuse-reset-with-12-volt-charge-pump
accompanying code and layout
https://github.com/telavivmakers/High-Voltage-ATtiny-ATTami-Fuse-Reset
TITLE: ATTiny Fuse Reset with 12V Charge Pump
Document Number:
REV:
Date: 2/23/2015 11:07:29 PM
Sheet: 1/1

Operation

- There are two modes of operation, each of them runs under different environment:
  - HV fuse reset mode
  - ISP mode
- There are two control buttons. (The control buttons functionality is the same under both modes of operation)
  - ATTami test button
  - ATTAMI load bootloader button

Operation cont.

- HV fuse reset mode
  - In this mode you can:
    - Reset the fuses to factory default state.
    - Set the fuses to a predefined state.
    - Define the predefined fuses state.
  - To get into the mode:
    - Connect the Arduino to the PC via a USB cable
    - Set the ISP-HV selector to HV. (HB LED should on)
    - Load Arduino IDE
    - Open the Serial monitor (if necessary set it to 19200 baud rate)
  - To reset the fuses
    - Follow the instructions on the screen menu (While the HV is applied, in options 1 or 2, you should notice a short blink on the HV LED.)

Operation cont.

- ISP programming mode
  - In this mode you can:
    - Upload a sketch into the ATtiny.
  - To get into the mode:
    - Connect the Arduino to the PC via a USB cable
    - Set the ISP-HV selector to ISB. (HB LED should pulse)
    - Load Arduino IDE
    - Set the board according to your ATtiny MCU
    - Set programmer to “Arduino as ISP
  - To program the ATtiny
    - Load your sketch into the IDE
    - Click on the “Upload” button. (error LED might blink at start and then turn off, pmode LED should blink while Uploading is performed and then turned off.)

Operation cont.

- ATTami test
  - Operates on both HV and ISP modes.
  - On Both modes when not actually programming the ATtiny VCC is off and the reset pin is LOW.
  - When the test pushbutton is pressed Vcc is applied to the ATtiny and the reset pin goes HIGH. (The ATTami, or Attiny, will commence running)
  - Watch the test LEDs for the ATP pattern.

Operation cont.

- ATTami load bootloader
  - Operates on both HV and ISP modes.
  - Operates even when the Arduino is not connected to a computer (5V source is nedded)
  - Momentary Push the “ATTami load bootloader” push buton.
  - The Arduino will upload the boot loader Hex file to the ATTami.

This function is not yet implemented

Software build

- The software comprises 3 ino files that are on the same directory
- The directory name is “HV_ISP_programmer”
- The 3 files are: “HV_ISP_programmer.ino”, “ATTinyFuseReset.ino” and “ArduinoISP_ATTami”.
- “HV_ISP_programmer.ino” contains the setup() and loop functions(), “ATTinyFuseReset.ino” contains the HV programming functions, “ArduinoISP_ATTami” contains the ISP functions.
- The IDE compile and build them together and Upload the produced Hex file to the Arduino. You should do it once before using the Programmer.