Building Interactive Devices and Objects

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

michael.rohs@ifi.lmu.de

MHCI Lab, LMU München

Today

- Course Overview & Project Topics
- Schedule & Organization
- Introduction (Hardware, AVR Eclipse)
- Exercise 1

Building Interactive Objects

Im Praktikum Building Interactive Objects geht es um den Entwurf und die Entwicklung interaktiver Geräte und Objekte. Dies umfasst die Komponenten Software, physisches Objekt/Gehäuse und eingebettete Elektronik. Die Grundlagen dazu werden in der Veranstaltung vermittelt. Die Teilnehmer erlernen die Entwicklung eingebetteter interaktiver Geräte mit Hilfe von Mikrocontrollern, das Bauen physischer Prototypen mit Lasercutter, 3D-Drucker und anderen Werkzeugen, sowie die zugrundeliegenden Interaktionskonzepte. Die Teilnehmer bauen in Gruppen interaktive Geräte und Objekte, programmieren interaktive Mini-Roboter, entwickeln steuerbare Pico-Projektoren, bauen Multi-Touch-Displays, sowie gestengesteuerte tragbare Geräte mit Tiefenkameras oder Beschleunigungssensoren. In einführenden Vorlesungen werden die notwendigen Grundlagen zu Mikrocontroller-Programmierung, Elektronik, Sensorik, Aktuatorik, Gestenerkennung und physical prototyping vermittelt.

TooMuch

PROJECTS

Project Topics

- Topic 1: Actuated Ambient Displays
- Topic 2: Animatronic

- You will work on one of these in groups of 4

Topic: Actuated Ambient Display

- Create an aesthetically pleasing object for the living room that mechanically visualizes information

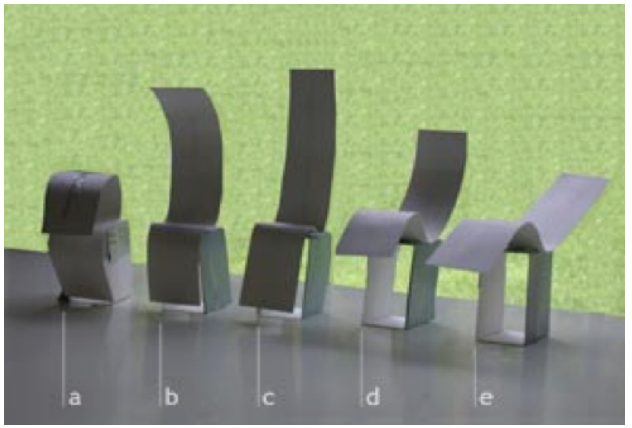
- (Lame) Example
  - Show temperature by moving analog scale to current value

Topic: Actuated Ambient Display

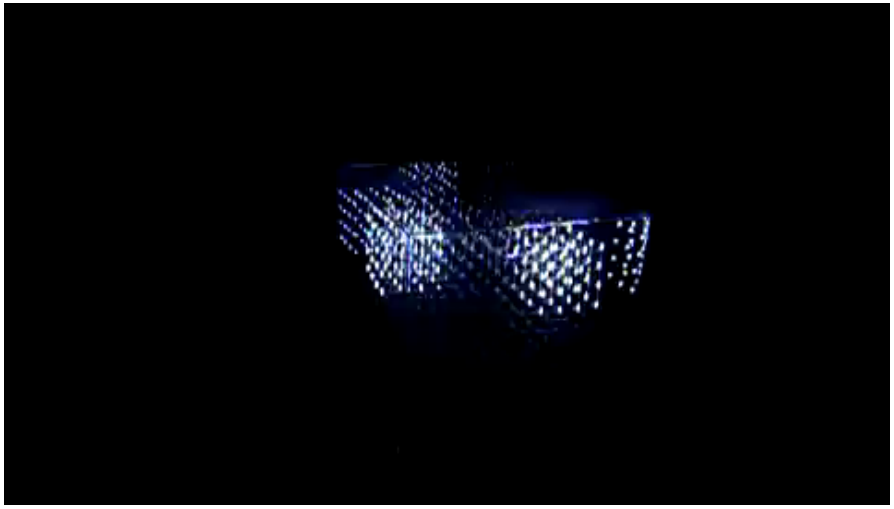
- Create an aesthetically pleasing object for the living room that mechanically visualizes information
- Pick an information source
  - Information from the Web
- Design the visualization
  - Selection: pick relevant data
  - Mapping: mechanically & visually encode the data
- “Actuated output”
  - Movement / position / orientation encodes data
  - No “traditional” 2D screen (only individual RGB LEDs)

Topic: Actuated Ambient Display

- Example: “Breakaway”
  - Sculpture lumps over if no break from desk work

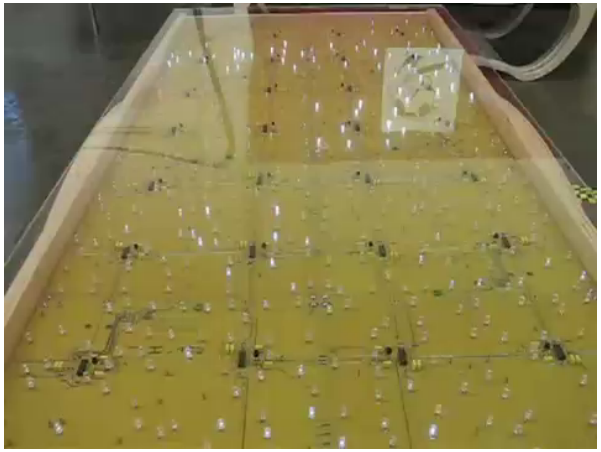


Jafarinaimi et al. Breakaway: An ambient display designed to change human behavior. CHI 2005 Extended Abstracts.



1000 LED Cube(James Clar, 2006)

Wooden Mirror (Daniel Rozin,



LED Coffee Table, Evil Mad Scientist (2007)

Visualization Design

- How to encode information mechanically?
- How to encode information with very few pixels?
- How to interact with the ambient display?

- Aesthetics?
- Mappings?
- Affordances?
- Conceptual models?

Information Sources

- Online information
  - News events, flight status (flightradar24), traffic, social media
- Awareness information
  - Number of people in office, energy usage, location of other person, mensa food options
- Natural phenomena
  - Weather, sunrise/sunset, tides, moon phase, vegetation periods, pollen count
- Physical activity
  - Equip person or animal with accelerometer, visualize physical activity, barometer/altimeter while hiking
  - Mobile phone to sense and send data

Actuation

- Self-contained physical artifact that can move
  - Communication via WLAN module
  - Processing on AVR microcontroller (ideally no PC required)
- Mounting
  - Placed on table, on floor, mounted on wall, on ceiling
- Output/actuation
  - Servo motors, RGB LEDs
- Input
  - Distance sensors, force sensors, touch pads, accelerometers, microphones

Topic: Animatronic

- Create an animated character / plush toy that can talk, make gestures, and can change facial expressions in response to user action
- Aspects
  - Usage scenario
  - Actuation of character
  - Audio output / projection output
  - Gesture / touch / speech input
- Example
  - Destini Disney

Disney Destini

http://www.youtube.com/watch?v=jZspn4r6ZMU

Topic: Animatronic

- Create convincing story / scenario
- Animate mouth, rotate head, body pose
- Recognize user distance
- Use audio localization of new Windows Kinect
- Use Windows Kinect speech recognition (Sven)
- Touch hand of animated character

Example Scenarios

- Studienberater für Schüler
  (welcher Studiengang passt zu mir?)
- Studienberater für Erstsemester
  (wie funktioniert studieren?)
- Fußballfan-Berater
  (welche Mannschaft passt zu mir?)
- Ticketberater für die MVV
  (welches Ticket, für Gruppe von
  3 Erwachsenen für 2 Tage München?)

System Aspects (both projects)

- Processing
  - Microcontrollers, I/O control, sampling
- Communication
  - WLAN modules
- Input
  - Switches, potentiometers
  - Distance sensors, pressure sensors
  - Accelerometers, gyroscopes
- Output (physical actuation, no rectangular display)
  - (RGB) LEDs
  - (Servo) Motors
  - Audio
  - Micro-projector (creative use, e.g., rain/snowfall to visualize weather)

Required Knowledge and Skills

- Develop concept
  - Interaction design, paper prototyping, scenario, persona
- Build physical object
  - Laser cutter, drill, 3D modeler
  - 3D modeling software (OpenSCAD)
- Build electronics
  - Programming microcontrollers (µCs, ADCs)
  - Basic electronics (resistors, capacitors, voltage controllers)
  - Combining components (µCs, LEDs, motors, WLAN modules)
- Evaluate object
  - Use yourself over several days
  - Write up findings, project blog on team activities

SCHEDULE

Schedule

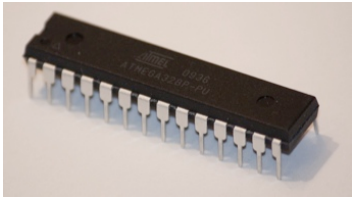
- Introduction phase
  - Tutorials / lectures
    - Form groups of 4 in first session, begin concept development
  - Presence exercises
    - Program microcontroller, control LEDs, control servos, read sensor data (potentiometer, distance sensor), capacitive sensing
- Project phase (blog)
  - Concept development (← begin early)
  - Concept presentation (2 minutes + discussion)
  - Implementation
  - Evaluation and write-up (2 weeks)
  - Presentation (7 minutes + discussion)

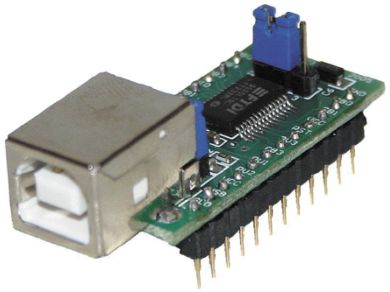
Schedule

#	Date	Topic	Group Activity
1	19.4.2012	Session 1: Introduction	Team building
2	26.4.2012	Session 2: Microcontrollers & Electronics	
3	3.5.2012	Session 3: Sensors	Concept development
4	10.5.2012	CHI	Concept development
5	17.5.2012	Christi Himmelfahrt	Concept development
6	24.5.2012	Session 4: Actuators	Concept presentation, Hardware requ.
7	31.5.2012	Session 5: Physical Objects (Sven)	
8	7.6.2012	Frohnleichnahm	Project
9	14.6.2012		Project
10	21.6.2012		Project
11	28.6.2012		Project
12	5.7.2012		Project
13	12.7.2012		Evaluation
14	19.7.2012		Evaluation, Presentation

Sessions 1: Introduction

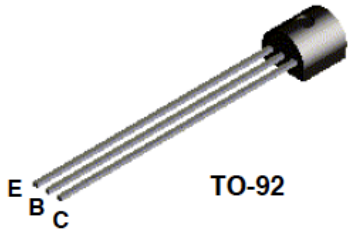
- Present topics, present schedule, organization, form groups, introduction to microcontrollers / AVR, how to read data sheets
- Exercises
  1. Install Eclipse-AVR
  2. Set up programmer board
  3. Breadboard, voltage controller, measure voltage & current





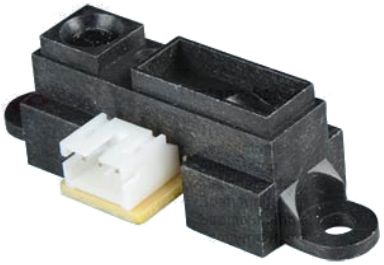
Sessions 2: Microcontrollers & Electronics

- More AVR, electronics basics, transistors, op-amps, using a multimeter / oscilloscopes
- Exercises
  1. Hello world (LED blinking)
  2. Connect USB breakout board
  3. Button debouncing, measure voltage, show bouncing
  4. Measuring currents and voltages of a transistor / op-amp



Sessions 3: Sensors, Concept Development

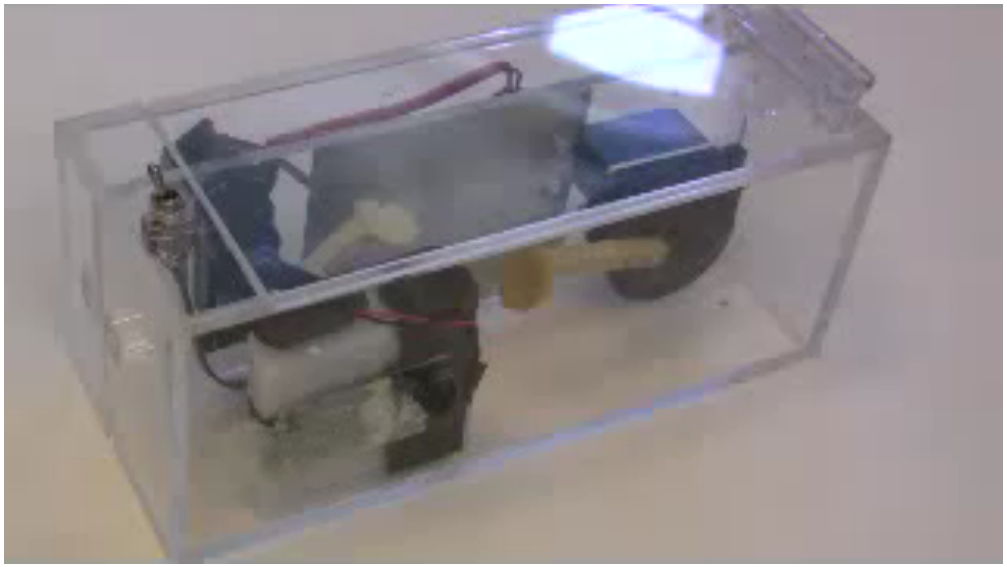
- LED mulitplexing, A/D-conversion, sensors
  - Potentiometer, capacitive, accelerometer, IR distance sensors
- Exercises
  1. Read potentiometer level to control LED blink rate
  2. Read IR distance sensor to control LED blink rate
  3. Concept development



Sharp GP2-1080 distance sensor

Sessions 4: Actuators, Concept Presentation, Hardware Requ.

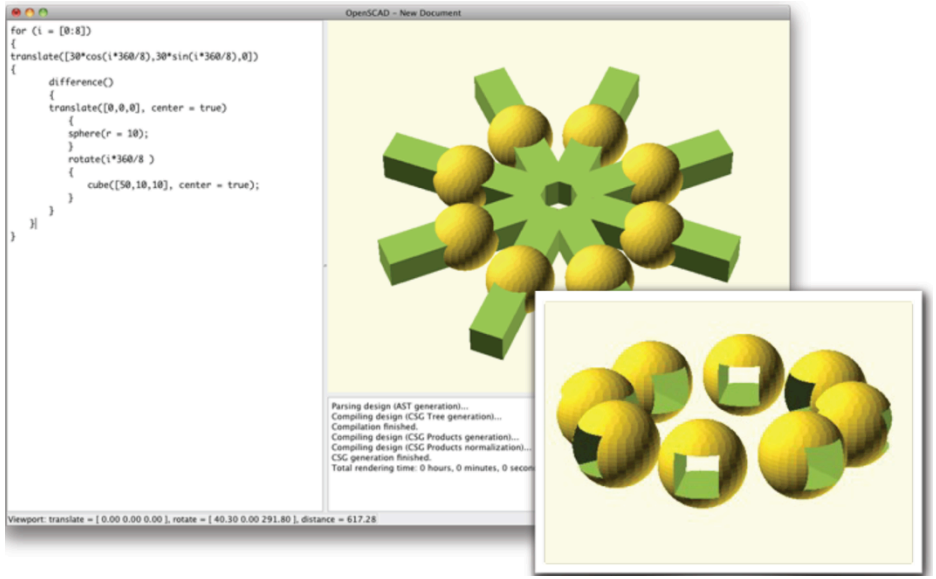
- Servo motors, DC motors, audio, PWM
- Exercises
  1. Control servo motor, attach pointer to motor and move to predefined positions, “the world’s most useless machine”
  2. H bridge for DC motor (Sven)



http://www.wimp.com/pointlessmachine/

Session 5: Physical Objects

- Building physical objects: 3D printing, modeling with OpenSCAD, later cutter usage, introduction to workshop
- Exercises
  1. Model a simple object and laser-cut it
  2. Create detailed list of additional hardware to order



Organization

- 4 SWS
  - Expect additional 1h per week for reading
- Weekly meetings
  - Thursday 16:00 (s.t.) – 20:00
  - Raum 107 oder Rückgebäude, Amalienstraße 17
- Homepage
  - http://www.medien.ifi.lmu.de/lehre/ss12/bio/
- Requirements
  - Must be present at every session
  - Grade based on individual exercises, participation in class, group project, final presentation
- Project budget (for additional hardware)

Teams

- Team 1
  - Yingding Wang
  - Moritz Bader
  - Cornelia Reithmeier
  - Saskia Friedich
  - Michael Marth
- Team 2
  - Felix Praschak
  - Fabius Steinberger
  - Frederik Brudy
  - Claudius Böttcher
- Team 3
  - Timo Becker
  - Marcel von Maltitz
  - Johannes Preis
  - ?
- Team 4 (lieber Thema 2)
  - Verena Lerch
  - Max Walther
  - Christiane Wölfel
  - Marion Koelle
  - M. Dadela

Team Organization

- Team blog
  - Project progress
  - Concept and scenarios
  - Design decisions
  - Photos, videos
- Team repository
  - Software and schematics
  - Git (http://git-scm.com/) or
  - Mercurial (http://mercurial.selenic.com/)
  - Hosting, for example: https://bitbucket.org, http://github.org
- Upload exercise solutions to UniWorX

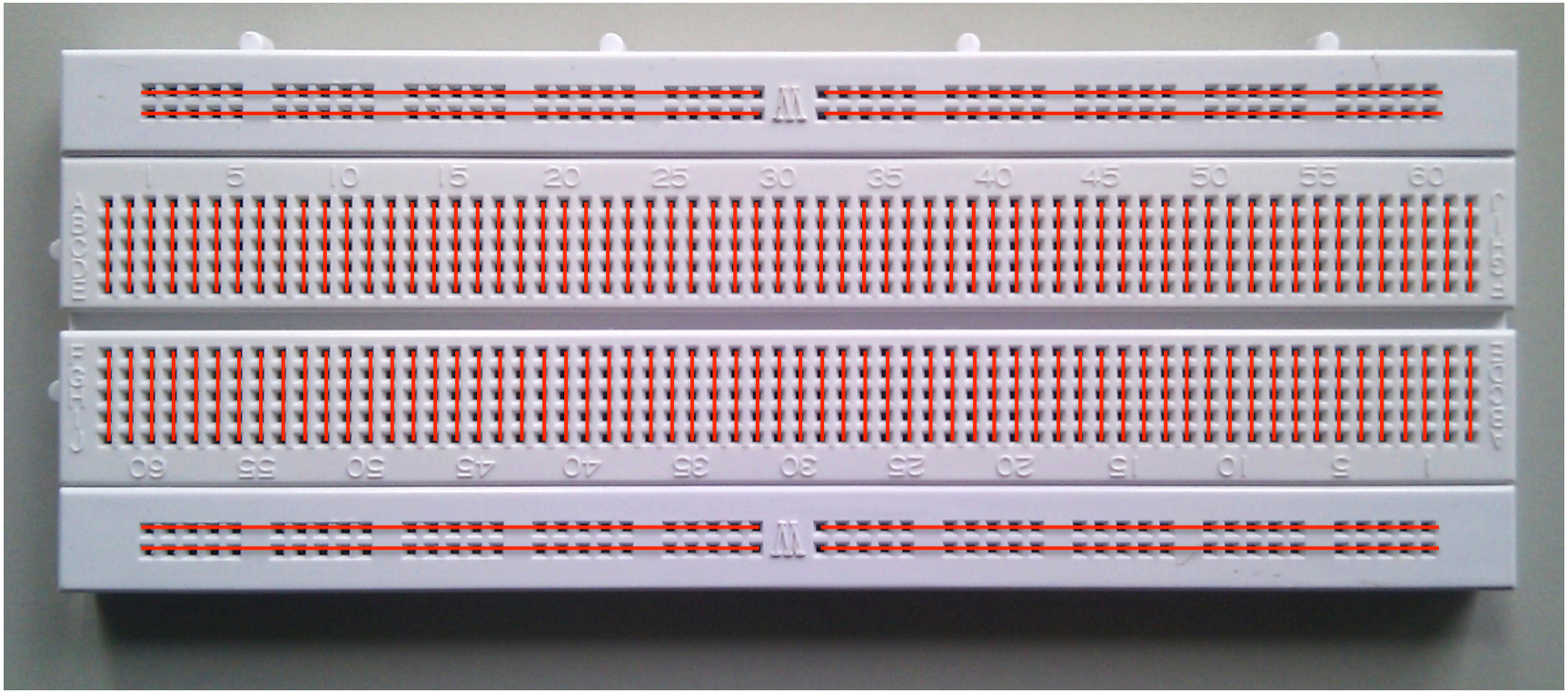
HARDWARE

Basic Hardware Kit

- Exercises (16 participants)
  - 2 Breadboards, AVR-Programmer, Programmer-Adapter, Netzteil, Multimeter, Seitenschneider, Drahtstücke (Set), Voltage regulator 5V, 3.3V, AVR ATtiny45, AVR ATmega8, LEDs, Widerstände (Set), Kondensatoren (Set Keramik, Set Elkos), Taster, Servomotor, USB-Serial Breakout board (FTDI), transistor, op-amp, 10kOhm Poti, IR distance sensor, FSR, accelerometer (analog), komprimierte Pappe (ähnlich MDF, Baumarkt, Sven), selbstklebende Kupferfolie
- Projects (4 groups)
  - RGB LEDs, Streifen
  - Roving WLAN Module

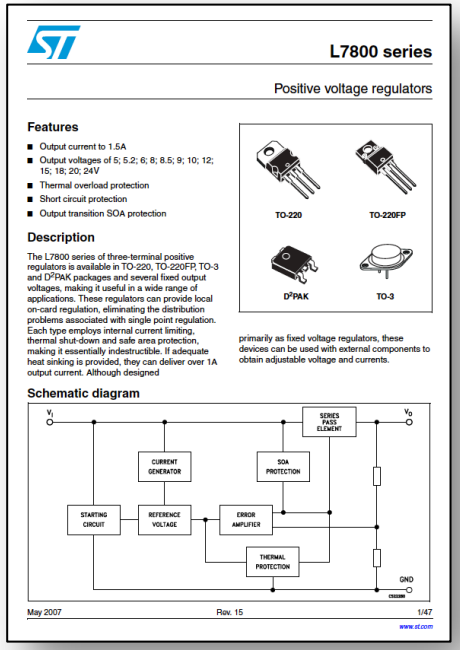
Breadboard

- Quick prototyping
  - Changing/adding components is easy
- Can get confusing soon (“spaghetti wires”)



Reading Data Sheets

- Extremely important to read carefully
  - Easy to find online
- Example: 7805 +5V voltage regulator
  - Operate according to “electrical characteristics”



L7800 series

Positive voltage regulators

Features

- Output current to 1.5A
- Output voltages of 5; 5.2; 6; 8; 8.5; 9; 10; 12; 15; 18; 20; 24V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection

TO-220 TO-220FP D²PAK TO-3

Description

The L7800 series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

Schematic diagram

May 2007 Rev. 15 1/47

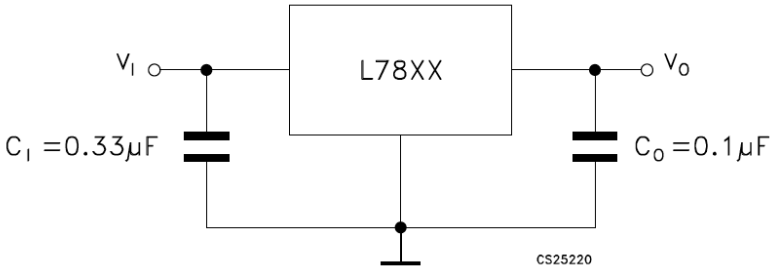
www.st.com

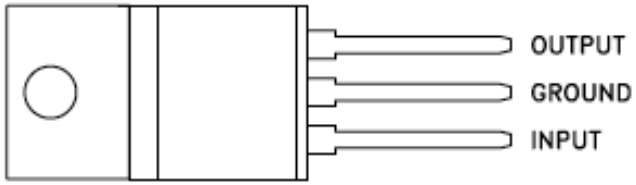
4 Electrical characteristics

Table 3. Electrical characteristics of L7805 (refer to the test circuits, T_J = -55 to 150°C, V_I = 10V, I_O = 500 mA, C_I = 0.33 µF, C_O = 0.1 µF unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	T_J = 25°C	4.8	5	5.2	V
V_O	Output voltage	I_O = 5mA to 1A, P_O ≤15W V_I = 8 to 20V	4.65	5	5.35	V

  - “Application Circuits” show typical usage

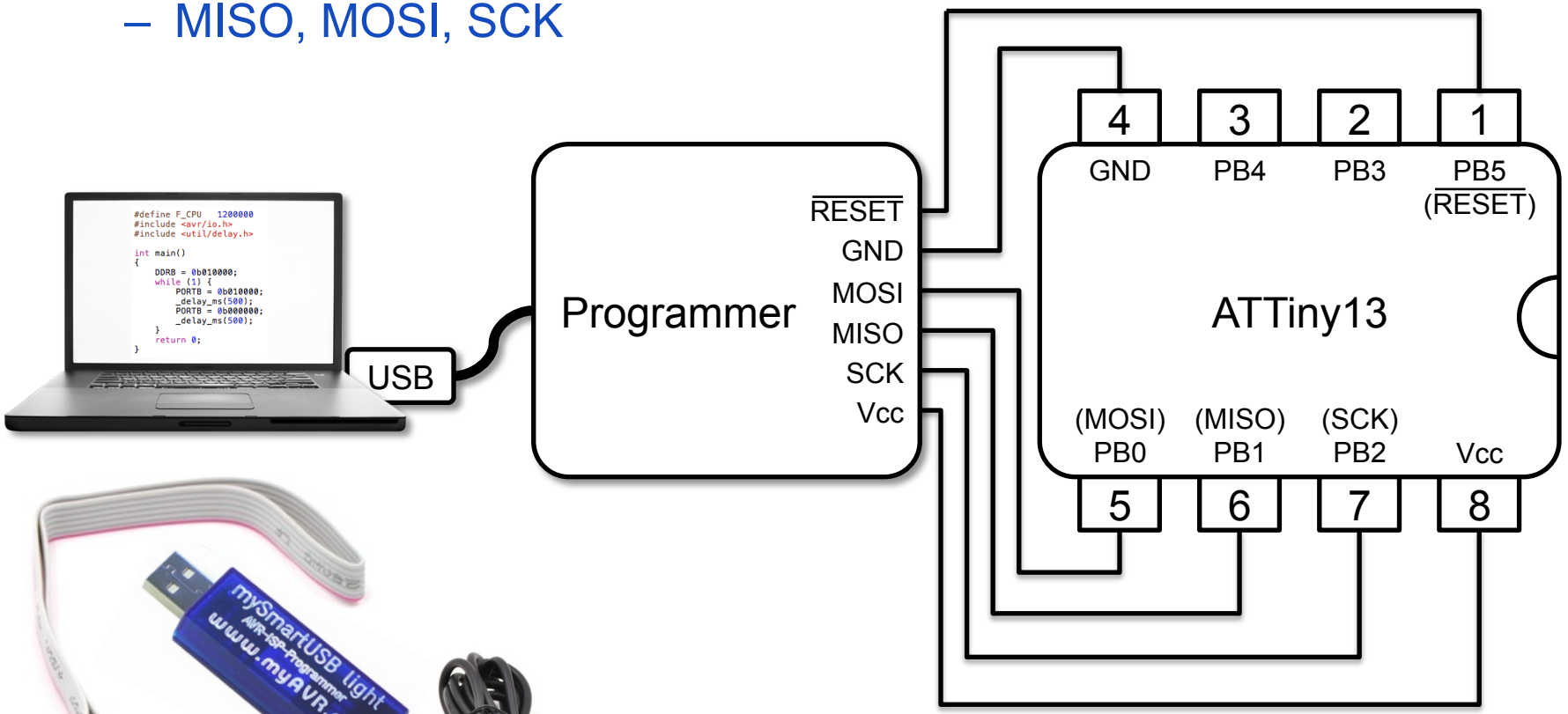




7805 +5V voltage regulator

Downloading the Program to the µC

- Serial programming via Serial Peripheral Interface (SPI)
  - MISO, MOSI, SCK



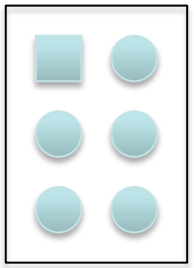
USB Drivers for “mySmartUSB light”

- USB chip CP2102 from Silicon Laboratories
- Windows

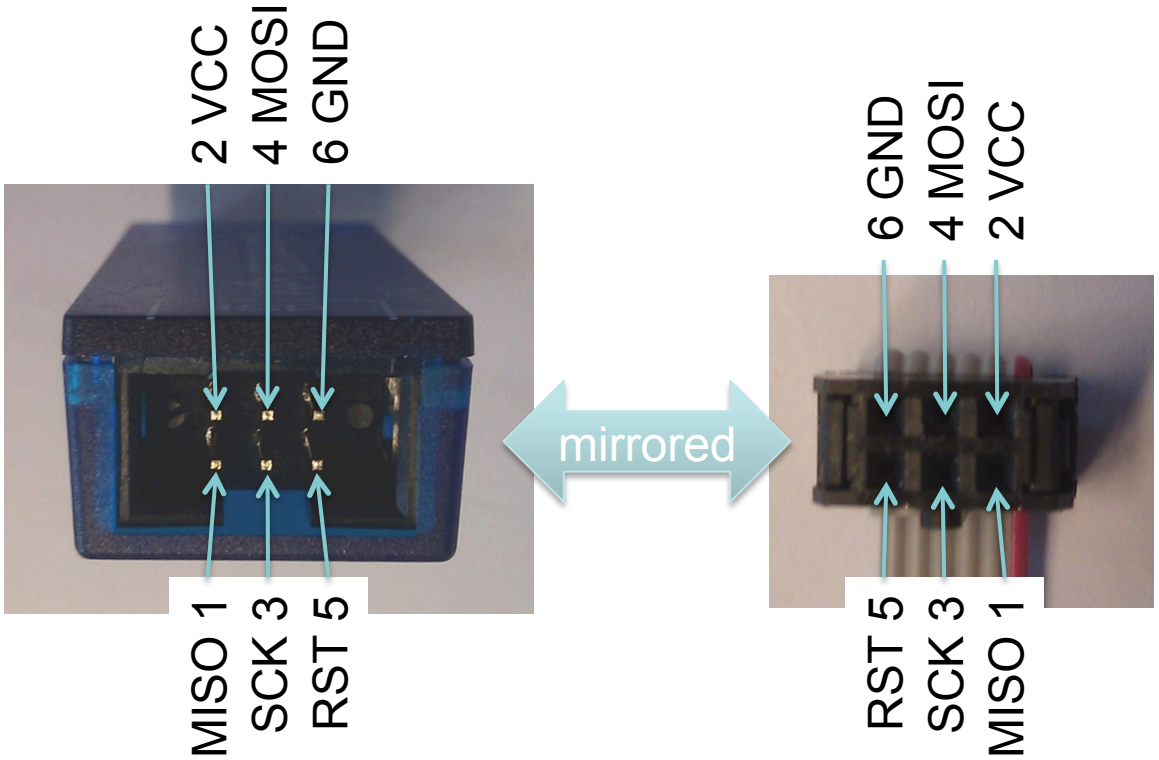
  http://shop.myavr.ch/index.php?sp=article.sp.php&artID=200006
- Mac OS X, Linux

  http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx

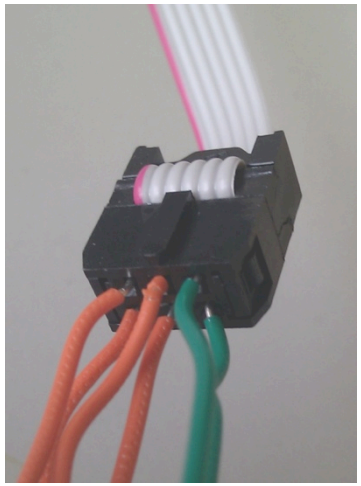
AVR ISP Connector



- Image of small PCB with one row of connecotrs
- http://itp.nyu.edu/physcomp/Tutorials/AVRCProgramming-Programmer



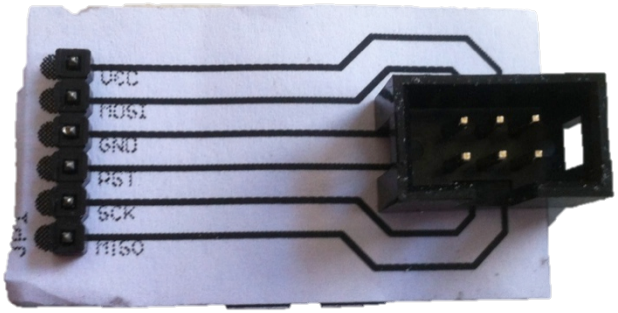
not recommended:

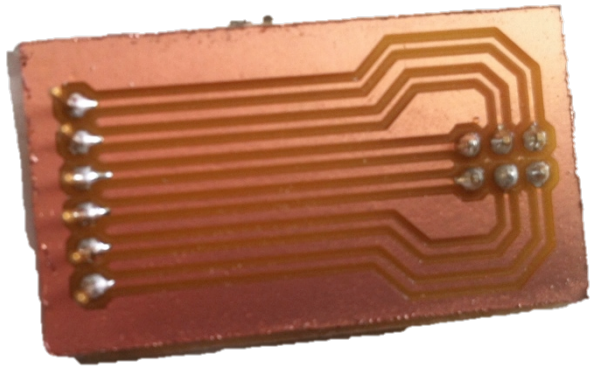


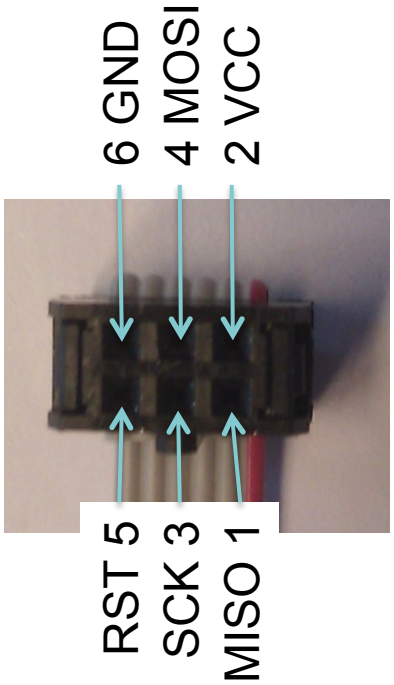
better solution:
solder small PCB
with 6x1 pins

SOLDERING

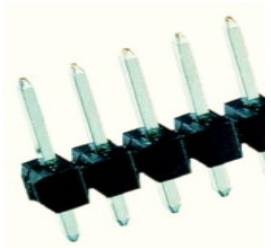
Mini Project: Programming Adapter





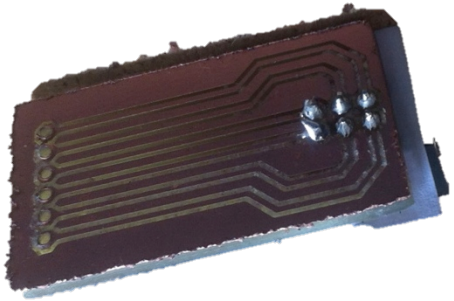


- Milled PCB (provided)
- 6-pin header for plugging into breadboard
- label for the six pins (2.54 mm spacing)
- two 3-pin headers for plug

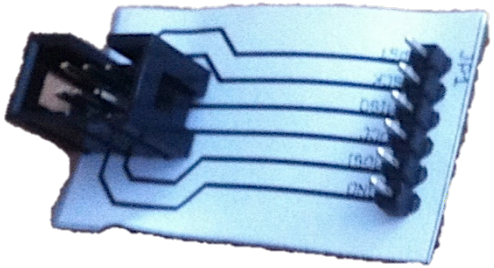


Assembling the Programming Adapter

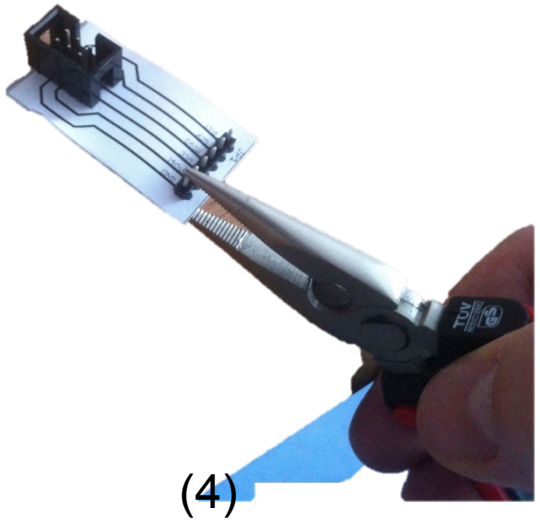
(1)



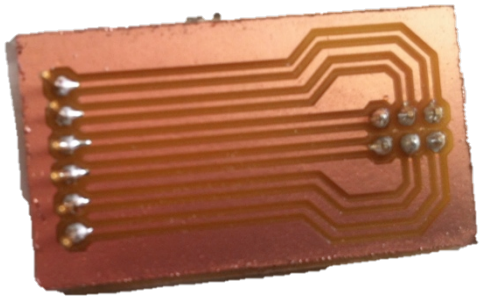
(2)



(3)

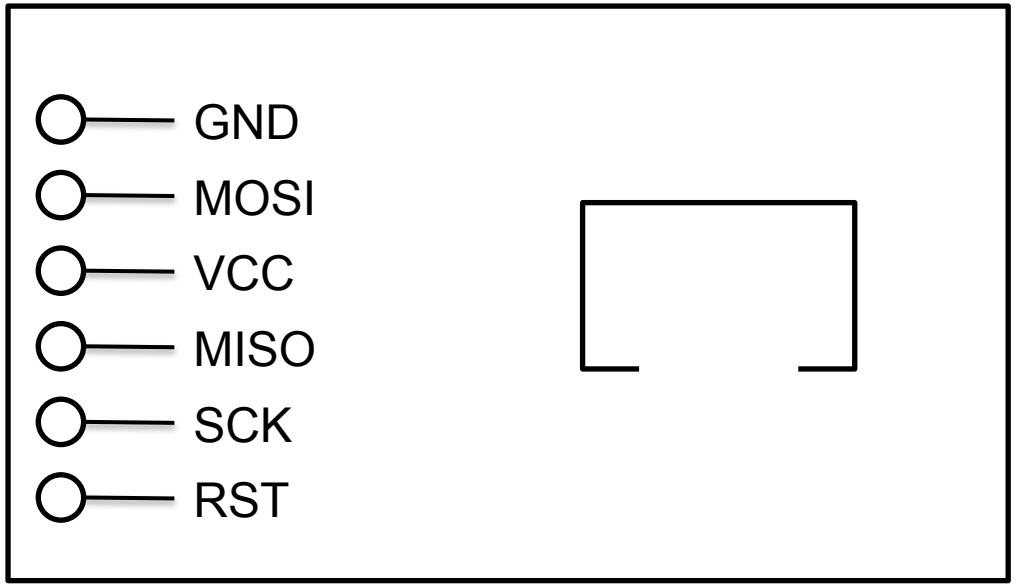


(4)



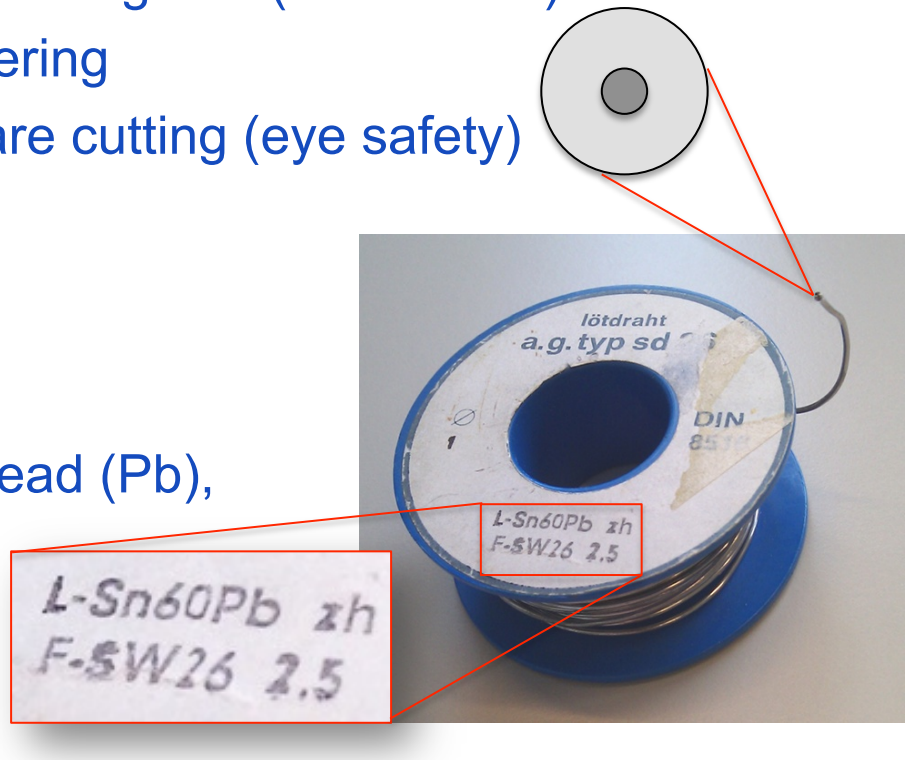
(5)

Programming Adapter: Pin Assignment



Soldering

- Consult “Soldering is easy – Here's how to do it”
  mightyohm.com/files/soldercomic/FullSolderComic_20110409.pdf
- Safety tips
  1. Don’t touch the tip of the soldering iron (300-380°C)
  2. Wash your hands after soldering
  3. Hold or cover the lead you are cutting (eye safety)
  4. Don’t breathe the smoke
- Solder is hollow and filled with flux (Flussmittel)
  - Example: 60% tin (Sn), 40% lead (Pb), flux: rosin (Kolophonium)
  - Lead-free solder in commercial applications



Soldering

- Place components close to printed circuit board (PCB), leads should be short
  - Long wires lead introduce capacitances
  - Cut leads after soldering, small wire cutter with flat edge
- Placing components
  - Pay attention to component polarity!
  - Bend out leads to keep component in place
  - Excessive heat destroys components
- Soldering iron
  - Heat up, wait, clean on wet sponge
  - Put back iron into stand at all times
  - Remove power plug when done

Soldering

- Soldering (quickly)
  - Press tip of soldering iron against lead and PCB contact pad (lead and pad need to heat up) for 2s
  - Add 1-3mm of solder (very little) where tip touches lead and PCB contact pad, pull solder away
  - Wait 1s, pull soldering iron away
- Control (and fix bad connections)
  - Control connections with a multimeter
  - A good solder connection has a flat profile
  - Bad solder connections can be fixed by repeating the process
  - Remove excess solder using a vacuum pump or copper braid

Good

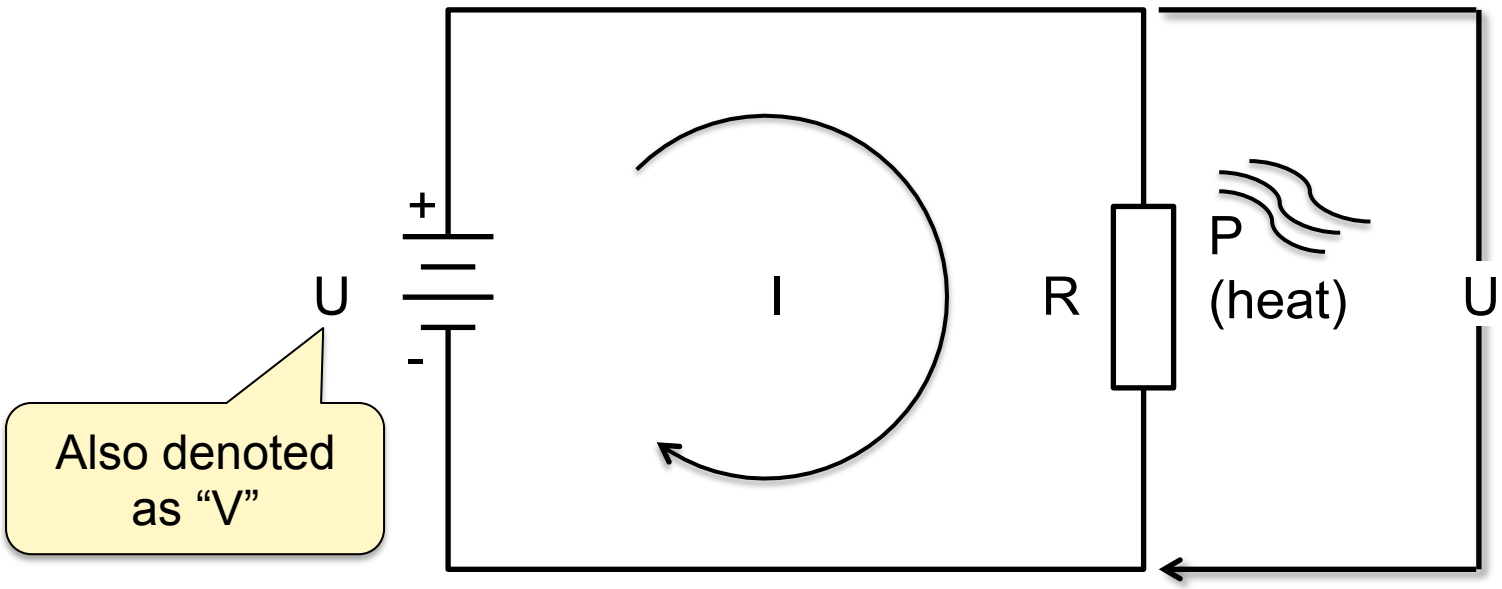
CC-BY-SA-3.0, GFDL

Bad

MEASURING VOLTAGE AND CURRENT

Ohm's Law

- Ohm's Law: $U = I\,R \Leftrightarrow I = U / R \Leftrightarrow R = U / I\ [\Omega]$
- Power loss: $P = U\,I \Leftrightarrow P = I^2\,R \Leftrightarrow P = U^2 / R\ [W]$
- Resistor power ratings



Resistors in Series

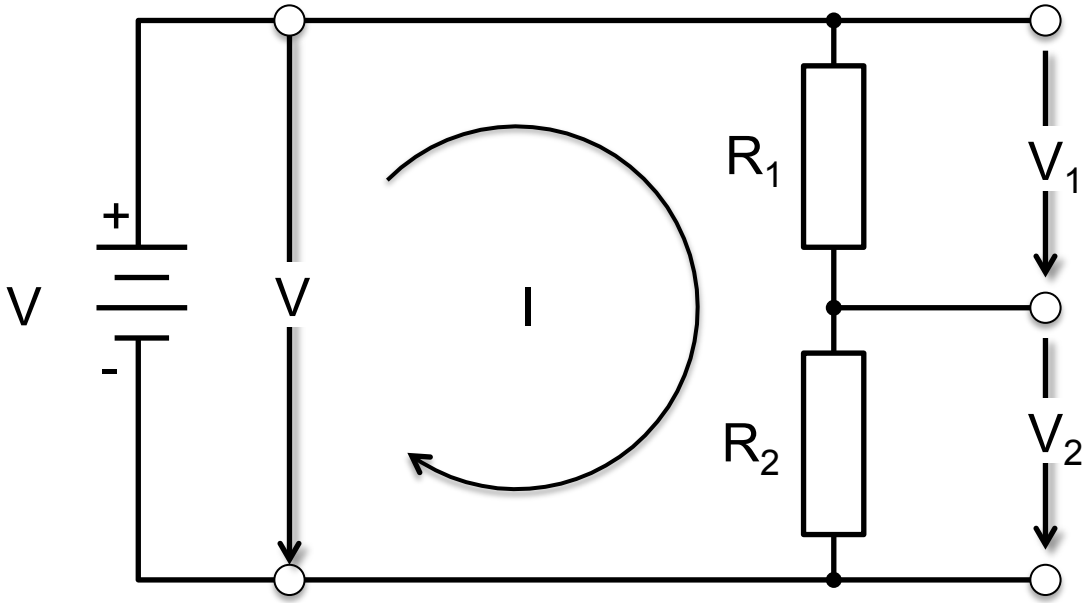
- V = V1 + V2
- R = R1 + R2
- Voltage divider

  $V_1 = V\, R_1 / (R_1 + R_2)$

  $V_2 = V\, R_2 / (R_1 + R_2)$

- Kirchhoff's Voltage Law

$$\sum_{closed\ loop} V_i = 0$$



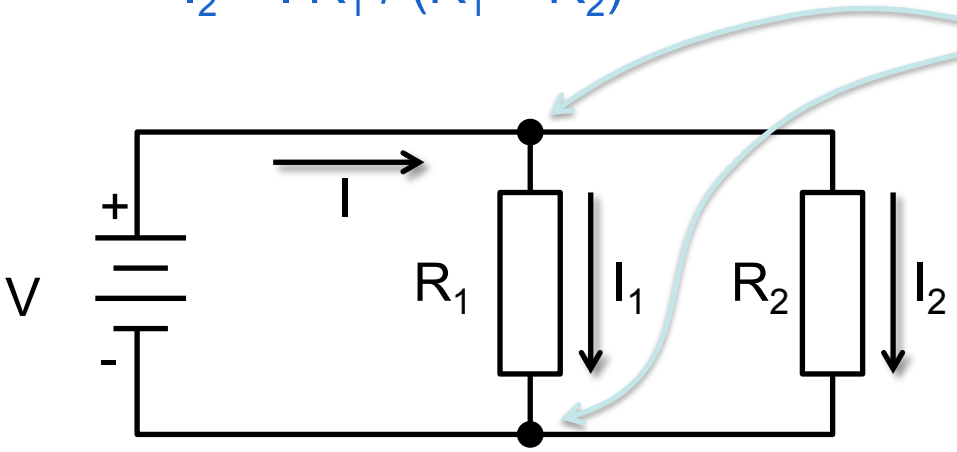
Resistors in Parallel

- $I = I_1 + I_2 \Leftrightarrow U / R = U / R_1 + U / R_2$
  $\Leftrightarrow 1 / R = 1 / R_1 + 1 / R_2$
  $\Leftrightarrow R = R_1 R_2 / (R1 + R2)$ (R: resistance)
  $\Leftrightarrow G = G_1 + G_2$ (G: conductance)

- Current divider
  - $I_1 = I R_2 / (R_1 + R_2)$
  - $I_2 = I R_1 / (R_1 + R_2)$

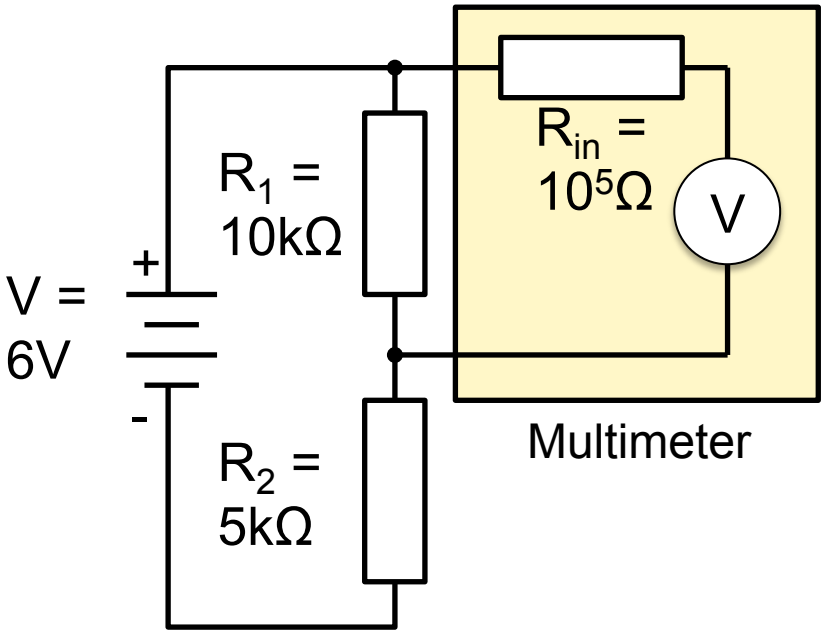
- Kirchhoff's Current Law

$$\sum_{node} I_i = 0$$



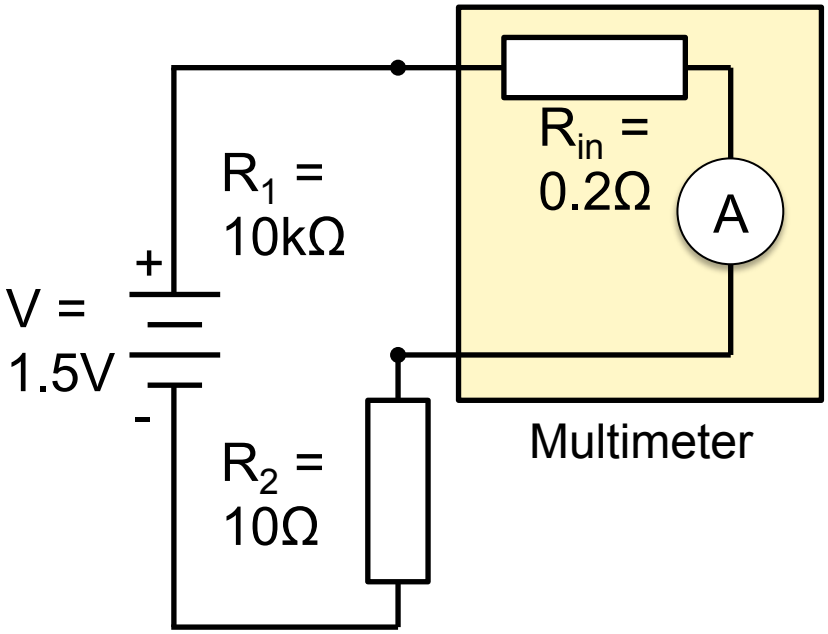
Measuring Voltage and Current

Measuring Voltage:



Actual voltage: 4.0V
Measured voltage: 3.9V
3% error

Measuring Current:

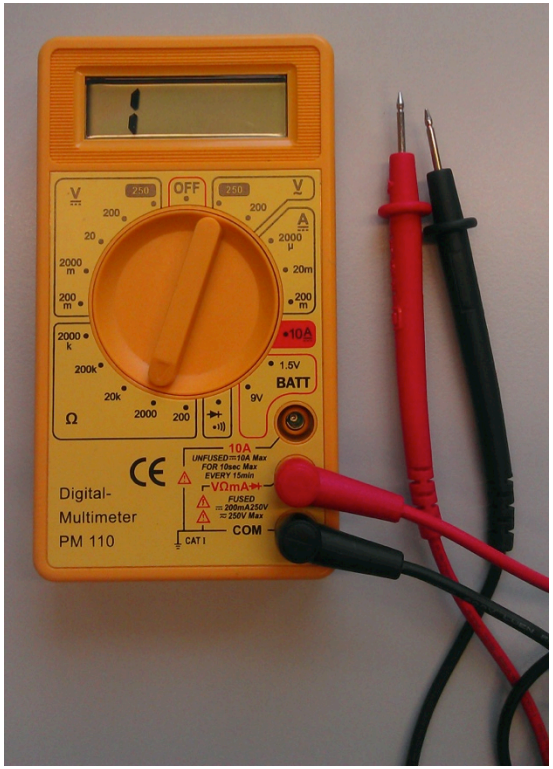


Actual current: 150mA
Measured voltage: 147mA
2% error

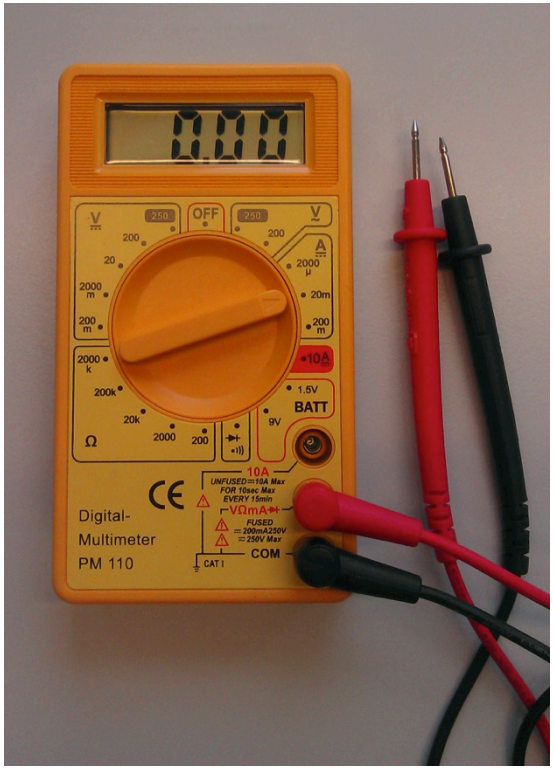
Multimeter

- Measure connectivity (“beep”), voltage, current, resistance

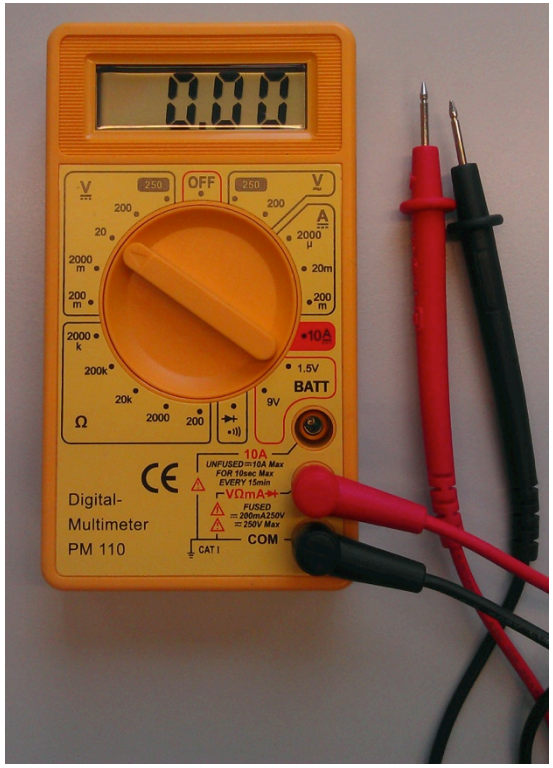
Resistance



Current



Voltage



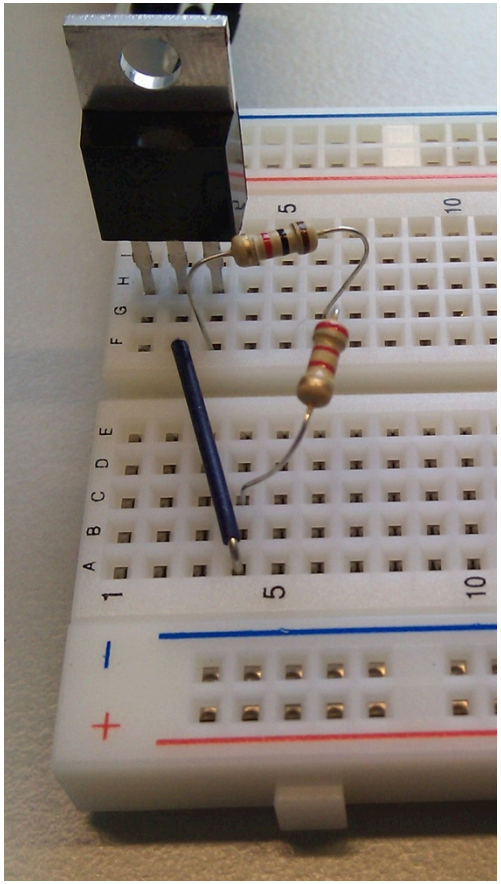
Resistor Color Codes

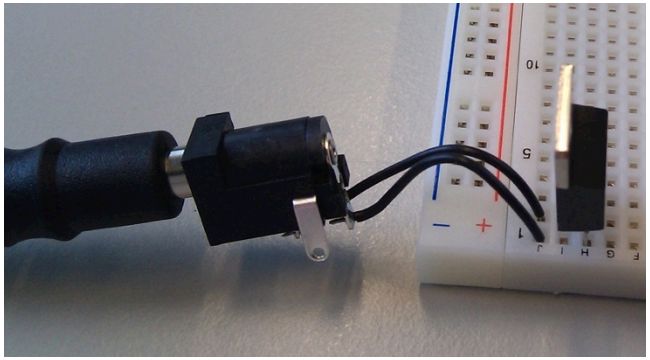
Farbe	1. Ring (1. Ziffer)	2. Ring (2. Ziffer)	3. Ring (3. Ziffer)	4. Ring (Multiplikator)	5. Ring (Toleranz)	6. Ring (Temp.-Koeffizient)
silber				10^{-2}		
gold				10^{-1}		
schwarz		0	0	10^{0}		200 10^{-6} K^{-1}
braun	1	1	1	10^{1}	±1 %	100 10^{-6} K^{-1}
rot	2	2	2	10^{2}	±2 %	50 10^{-6} K^{-1}
orange	3	3	3	10^{3}		15 10^{-6} K^{-1}
gelb	4	4	4	10^{4}		25 10^{-6} K^{-1}
grün	5	5	5	10^{5}	±0,5 %	
blau	6	6	6	10^{6}	±0,25 %	10 10^{-6} K^{-1}
violett	7	7	7		±0,1 %	5 10^{-6} K^{-1}
grau	8	8	8		±0,05 %	
weiß	9	9	9			

Source: Wikipedia

Measure Voltage and Current

- Use 7805 to create stable +5V voltage
  - Attention: 7805 will break if connected if the polarity is wrong
- Create a voltage divider using two resistors (R_1 = 1kΩ, R_2 = 2.2kΩ)
- Measure resistance of R_1 and R_2
  - Compare with encoded values
- Measure voltage across R_1 and across R_2
  - Check whether voltage conforms to voltage divider law
- Measure current
  - Check whether current conforms to Ohm's law





AVR ECLIPSE PLUGIN

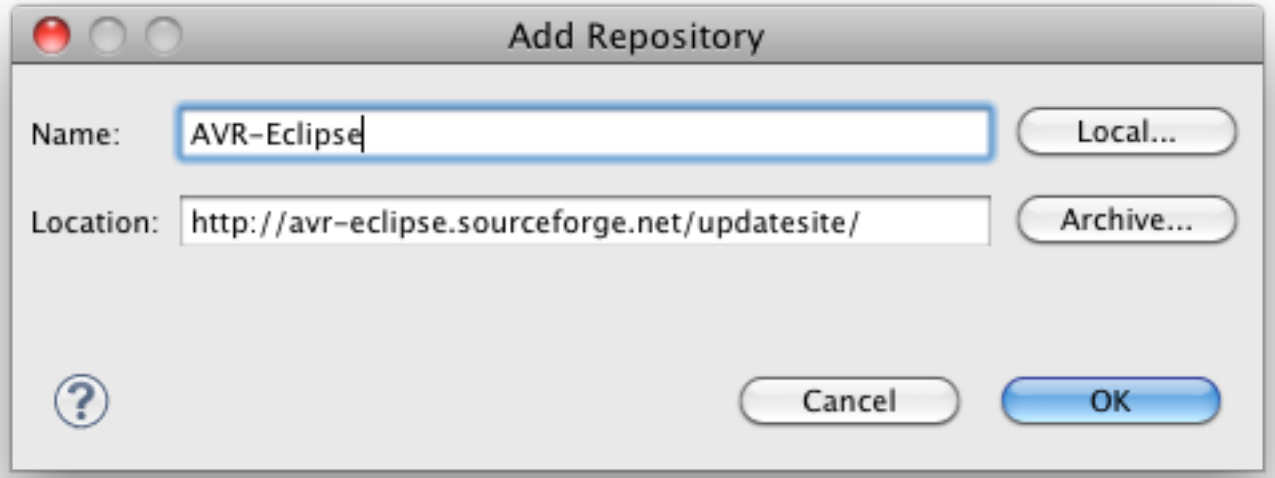
AVR Eclipse Plugin

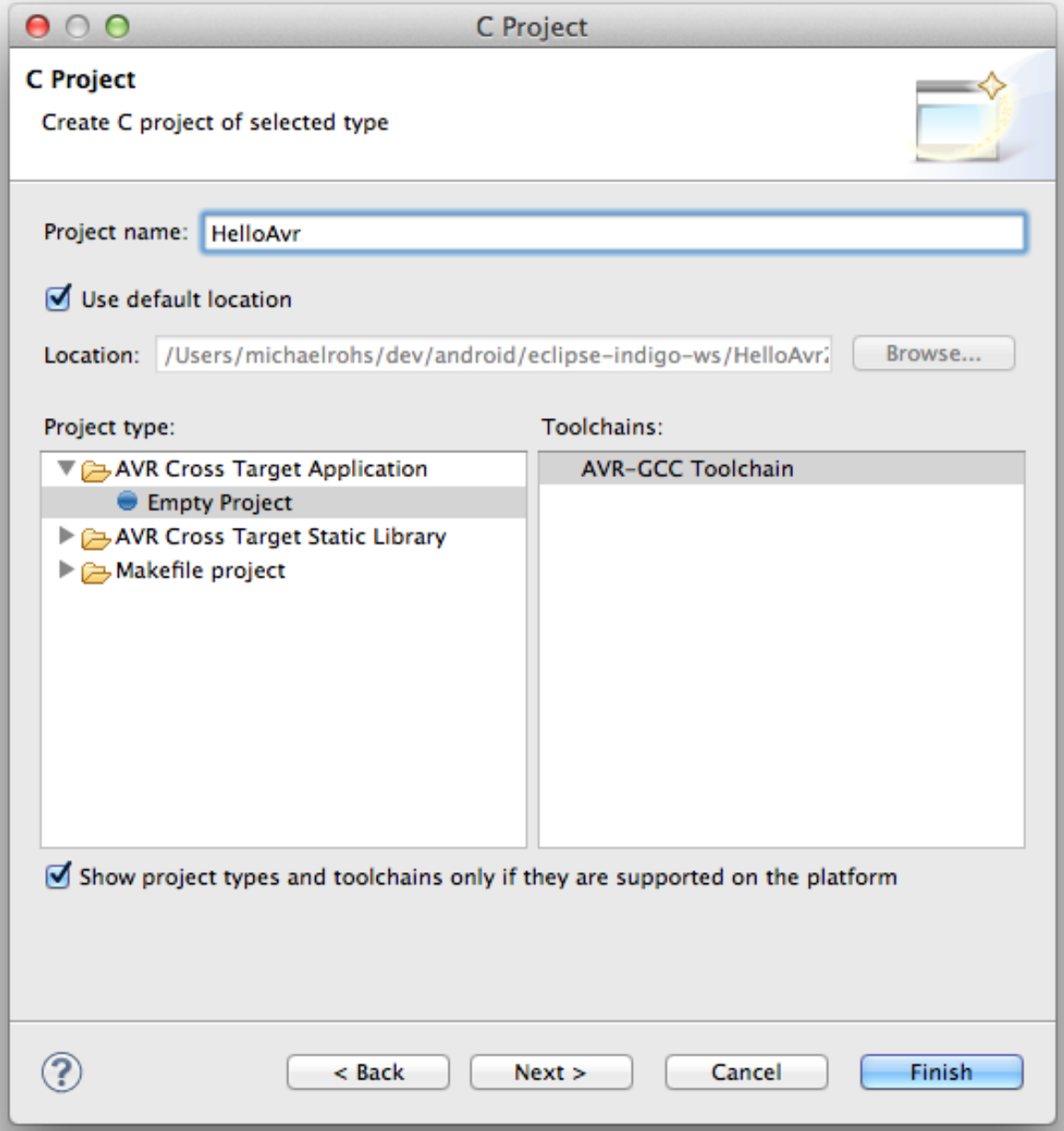
- http://avr-eclipse.sourceforge.net
- Installation (from within Eclipse)
  - Help | Install New Software… | Add…
  - http://avr-eclipse.sourceforge.net/updatesite
- Documentation
  - http://avr-eclipse.sourceforge.net/user%20manual/gettingstarted/gs_tutorials.html
- Create project
  - File | New | Project… | C/C++ | C Project | AVR Cross Target Application
  - Advanced settings… to specify target processor

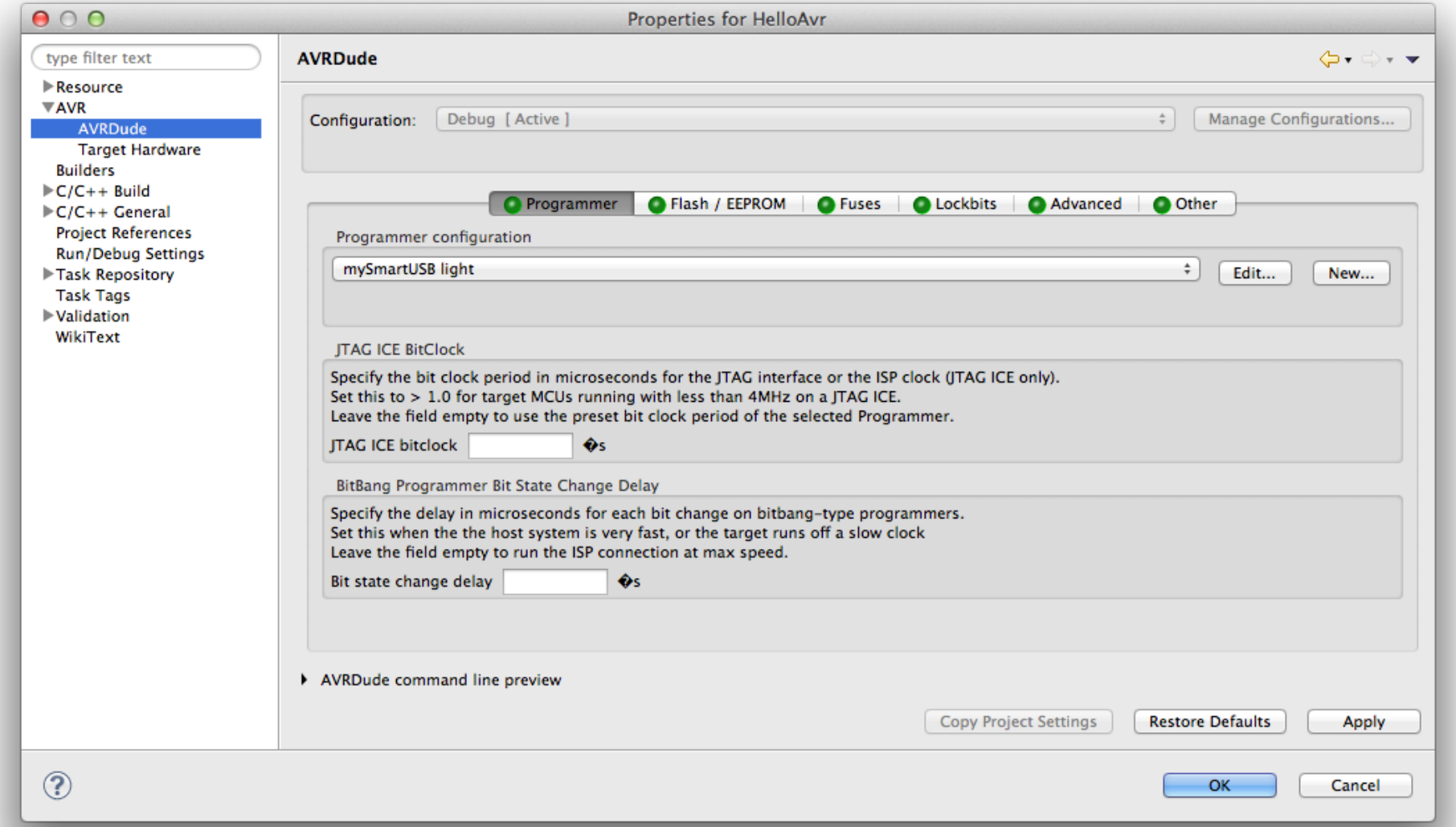
AVR Eclipse Plugin – Advanced Settings

- Target Hardware: Specify target microcontroller
  - MCU Type: ATtiny45 (later will also use ATmega8)
  - MCU Clock Frequency: typical values are 1 MHz (internal), 8 MHz (external quartz) , 16 MHz (external quartz)
- AVRDude: Install program on microcontroller
  - Programmer: Atmel STK500 Version 2.x firmware
  - Override default port: /dev/tty.usbserial-A100OXPZ

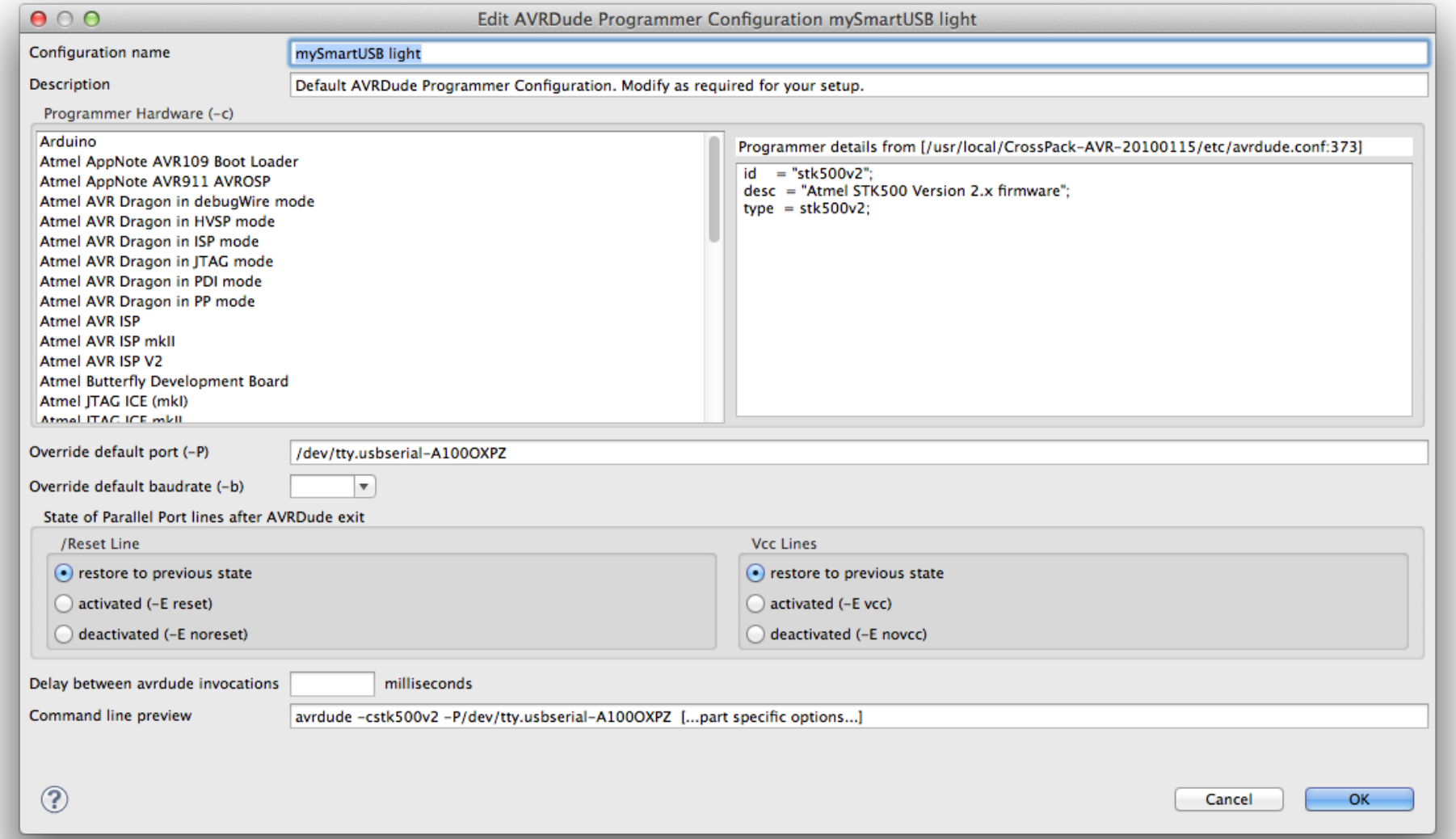
Install AVR Eclipse Plugin (from within Eclipse)



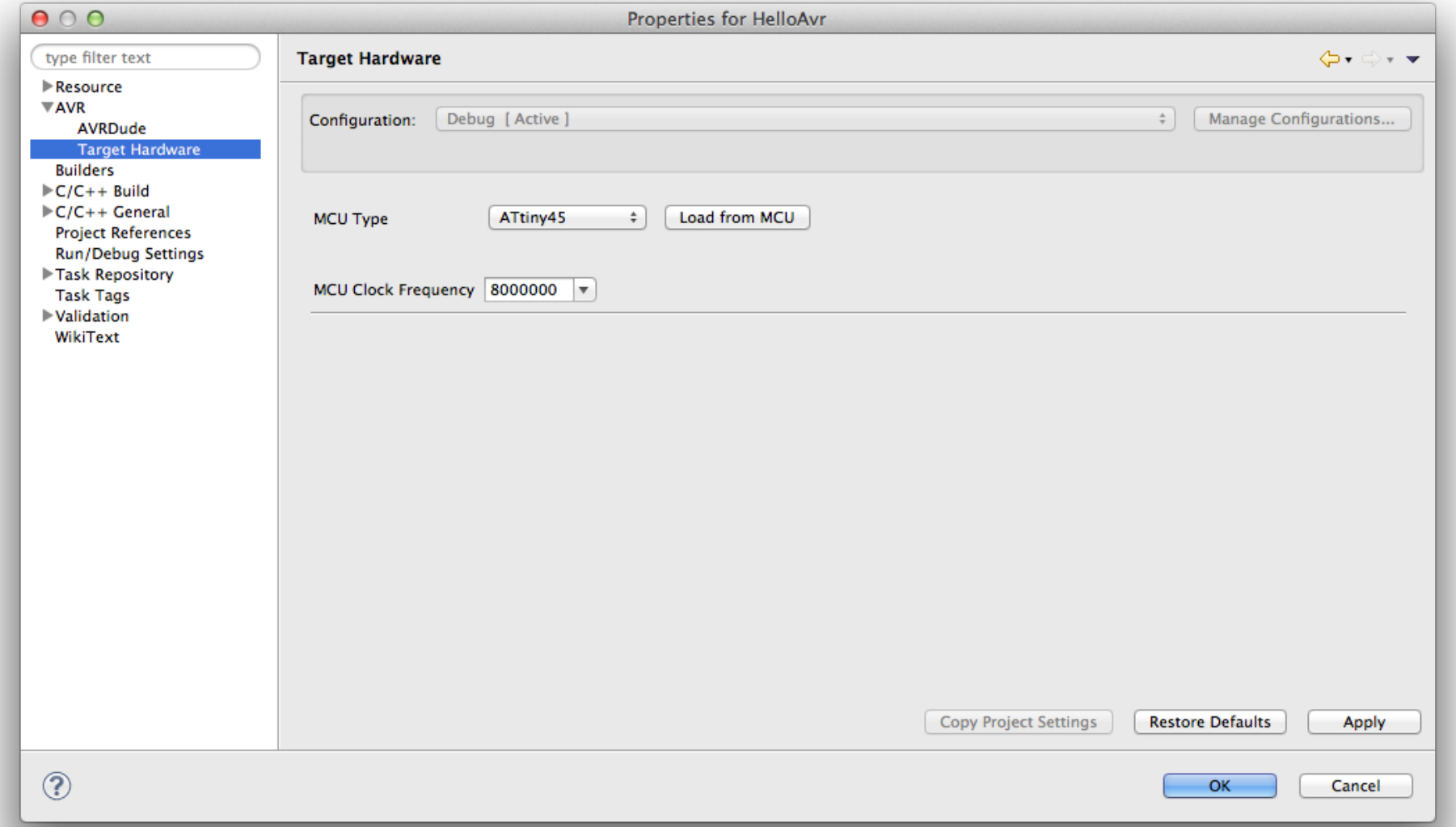

C Project
C Project
Create C project of selected type
Project name:
HelloAvr
Use default location
Location:
/Users/michaelrohs/dev/android/eclipse-indigo-ws/HelloAvr
Browse...
Project type:
AVR Cross Target Application
Empty Project
AVR Cross Target Static Library
Makefile project
Toolchains:
AVR-GCC Toolchain
Show project types and toolchains only if they are supported on the platform
< Back
Next >
Cancel
Finish


Properties for HelloAvr
type filter text
Resource
AVR
AVRDude
Target Hardware
Builders
C/C++ Build
C/C++ General
Project References
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText
AVRDude
Configuration:
Debug [Active]
Manage Configurations...
Programmer
Flash / EEPROM
Fuses
Lockbits
Advanced
Other
Programmer configuration
mySmartUSB light
Edit...
New...
JTAG ICE BitClock
Specify the bit clock period in microseconds for the JTAG interface or the ISP clock (JTAG ICE only).
Set this to > 1.0 for target MCUs running with less than 4MHz on a JTAG ICE.
Leave the field empty to use the preset bit clock period of the selected Programmer.
JTAG ICE bitclock
µs
BitBang Programmer Bit State Change Delay
Specify the delay in microseconds for each bit change on bitbang-type programmers.
Set this when the the host system is very fast, or the target runs off a slow clock
Leave the field empty to run the ISP connection at max speed.
Bit state change delay
µs
AVRDude command line preview
Copy Project Settings
Restore Defaults
Apply
OK
Cancel

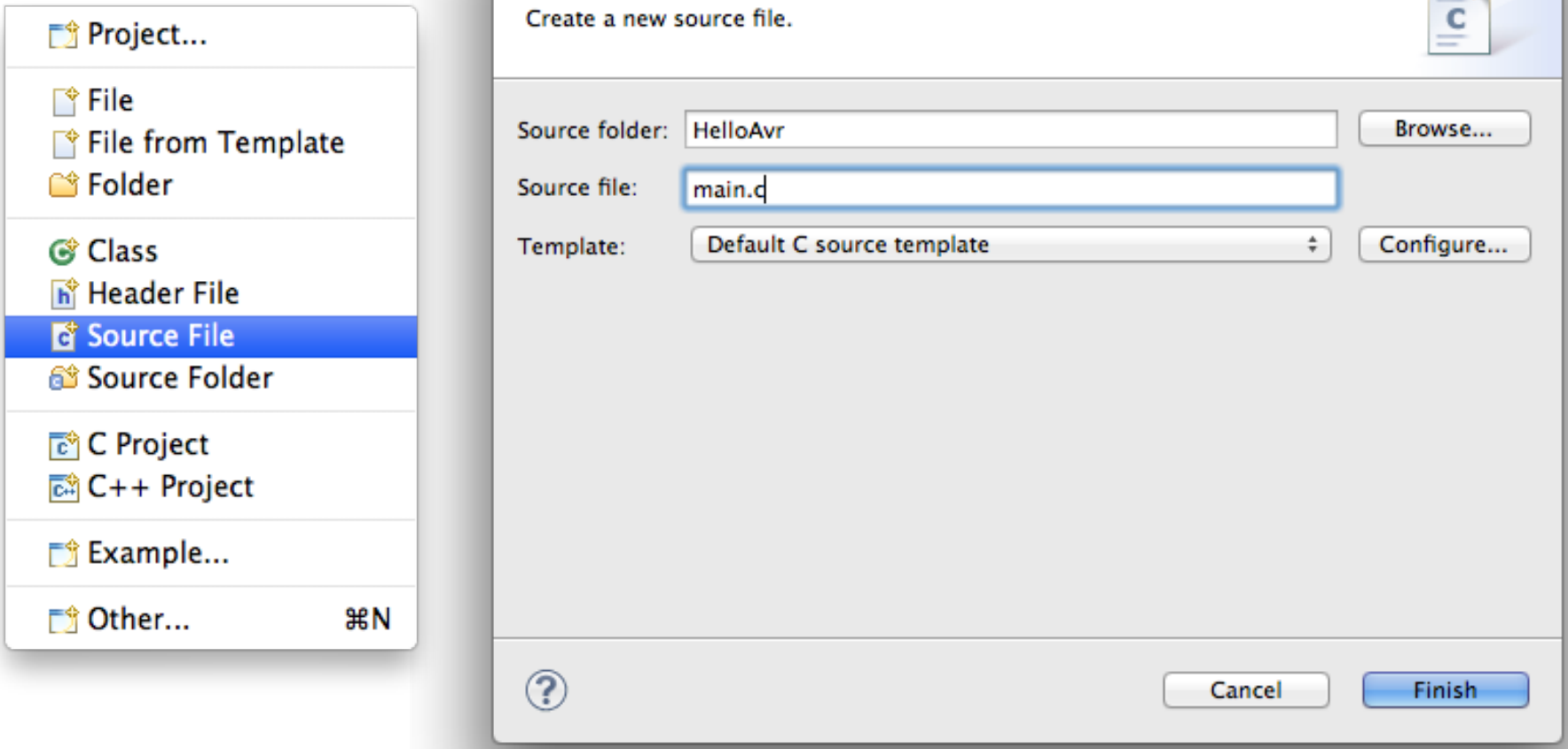
Programmer


Edit AVRDude Programmer Configuration mySmartUSB light
Configuration name
mySmartUSB light
Description
Default AVRDude Programmer Configuration. Modify as required for your setup.
Programmer Hardware (-c)
Arduino
Atmel AppNote AVR109 Boot Loader
Atmel AppNote AVR911 AVROSP
Atmel AVR Dragon in debugWire mode
Atmel AVR Dragon in HVSP mode
Atmel AVR Dragon in ISP mode
Atmel AVR Dragon in JTAG mode
Atmel AVR Dragon in PDI mode
Atmel AVR Dragon in PP mode
Atmel AVR ISP
Atmel AVR ISP mkII
Atmel AVR ISP V2
Atmel Butterfly Development Board
Atmel JTAG ICE (mkI)
Programmer details from [/usr/local/CrossPack-AVR-20100115/etc/avrdude.conf:373]
id = "stk500v2";
desc = "Atmel STK500 Version 2.x firmware";
type = stk500v2;
Override default port (-P)
/dev/tty.usbserial-A100OXPZ
Override default baudrate (-b)
State of Parallel Port lines after AVRDude exit
/Reset Line
restore to previous state
activated (-E reset)
deactivated (-E noreset)
Vcc Lines
restore to previous state
activated (-E vcc)
deactivated (-E novcc)
Delay between avrdude invocations
milliseconds
Command line preview
avrdude -cstk500v2 -P/dev/tty.usbserial-A100OXPZ [...part specific options...]
Cancel
OK

Target Hardware



New C Source File



Building the Project