



Pacemaker Challenge on AVR/Butterfly Boards

CIS 541: Embedded and Cyber Physical Systems
Spring 2010

Shaohui Wang
shaohui@seas.upenn.edu

Outline

- Project Overview
- Implementation Scheme
- Introduction to AVR/Butterfly Programming
 - Requirement
 - Setting Up Environment
 - Programming
 - Resources
- Project Logistics



PROJECT OVERVIEW

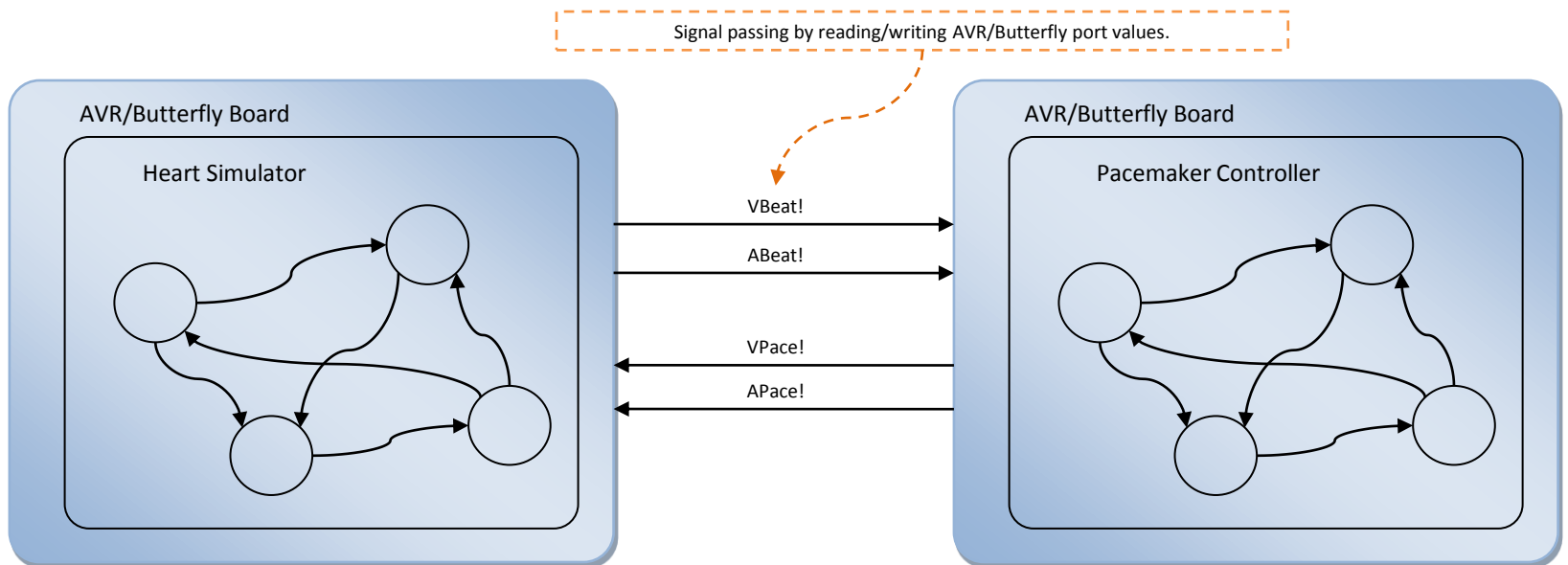
Project Overview

- Model based development of safety-critical embedded software
 - Create formal models
 - Verify properties
 - Code generation
 - Validation
- Main concerns
 - Embedded development
 - Timing constraints



IMPLEMENTATION SCHEME

Implementation Scheme



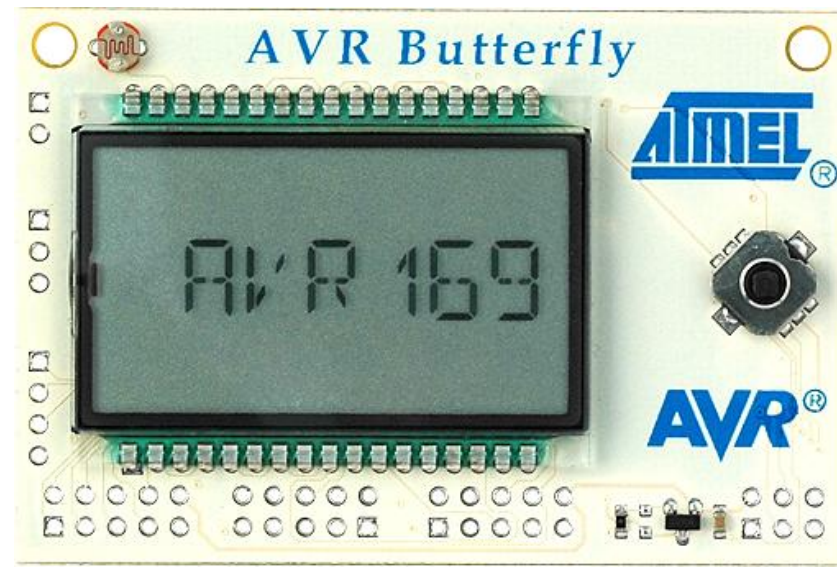
Heart Simulator

- Two behaviors
 - Beat by itself randomly in a range of pre-configured parameters
 - E.g., range=(100ms, 1900ms) means after a (ventricular) pace or beat, the next projected heart beat is randomly in this range.
 - Once paced by the controller, reset timer
 - May respond to controller, or may not, depending on your assumptions of the model
- Extensions
 - Cover both ventricle and atrium

Pacemaker Controller

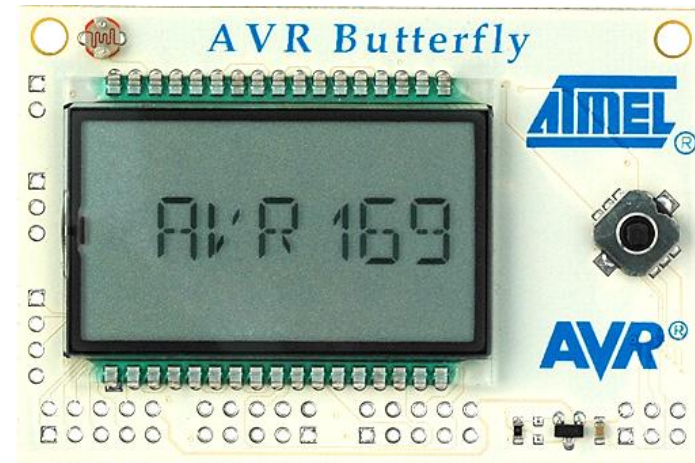
- Basic behavior
 - Monitor the heart by receiving signals
 - Pace the heart at designated time if the heart fails to beat by itself
- Extensions
 - DDD mode
 - Hysteresis pacing
 - Rate smoothing
 - (open ended)

THE AVR/BUTTERFLY BOARD



AVR/Butterfly Boards

- Features
 - ATmega169 AVR microcontroller
 - 100 segment LCD Display
 - 4Mbit Dataflash
 - 32kHz oscillator for RTC
 - 4-way directional button
 - Light sensor (LDR)
 - Temperature Sensor (NTC)
 - Speaker for Sound Generation
 - Access to peripherals through header connectors
 - RS-232 Level Converter





PROGRAMMING

Setting Up the Environment

- AVR Studio 4.18 & Service Pack 1
 - Easy to use IDE
 - Programming
 - Debugging
 - Dumping .hex files to AVR/Butterfly board
 - <http://www.atmel.com/dyn/products/avrstudio>
- WinAVR
 - Comprehensive Toolchain
 - Provide GCC Port – use C to program AVR/Butterfly
 - <http://sourceforge.net/projects/winavr/files/>
- Installation
 - AVR Studio & SPI first, then WinAVR
 - Automatically set up AVR Studio for creating C projects

The bf_gcc Sample Program

- Overview
 - A port of the on board AVR/Butterfly program to GCC
 - Contains examples of all features of AVR/Butterfly
 - Good starting point for programming
 - Available online
 - Newest 12/17/2009
 - http://www.siwawi.arubi.uni-kl.de/avr_projects/bf_gcc_20091217.zip

Main Logic for Sample Program

- Initialization (pStateFunc, PowerSave, clocks, etc)
- Main loop
 - If not in PowerSave mode
 - Print state indication text
 - Get an input from the button
 - Execution code associated in the current state
 - If input from the button is not NULL, transit to next state
 - Update state function pointer pStateFunc
 - Code for AutoPowerSave mode
 - If system idle time is larger than preset, set PowerSave to TRUE
 - Code for AutoPress mode
 - If a button is held for some time, consider it pressed even if released, until changed
 - Code for SleepMode
 - Turn off LCD
 - If UP button is pressed, return to normal mode

Programming with AVR/Butterfly Features – Joystick

- Use `input=getkey()` to get input, and compare with predefined values
 - `KEY_ENTER(·)`
 - `KEY_NEXT(→)`
 - `KEY_PREV(←)`
 - `KEY_PLUS(↑)`
 - `KEY_MINUS(↓)`

Programming with AVR/Butterfly Features – LCD

- Use functions defined in LCD_functions.h
 - void `LCD_puts_f`(const char *pFlashStr, char scrollmode);
 - Writes a string stored in flash to the LCD
 - void `LCD_puts`(char *pStr, char scrollmode);
 - Writes a string to the LCD
 - void `LCD_UpdateRequired`(char update, char scrollmode);
 - Tells the LCD that there is new data to be presented
 - void `LCD_putc`(uint8_t digit, char character);
 - Writes a character to the LCD
 - void `LCD_Clear`(void);
 - Clear the LCD
 - void `LCD_Colon`(char show);
 - Enable/disable colons on the LCD

Programming with AVR/Butterfly Features – I/O of Ports

- Two steps
 - Set the data direction for each pin of the port
 - E.g., `DDRB=0xF0`
 - set pins 0 to 3 (lower) of PORTB as input and
 - pins 4 to 7 (upper) as output
 - I/O
 - Read values of PINB for values of Port B
 - Write values to PORTB for Port B
- Similar for Port D
 - Note: there are only limited pins for use

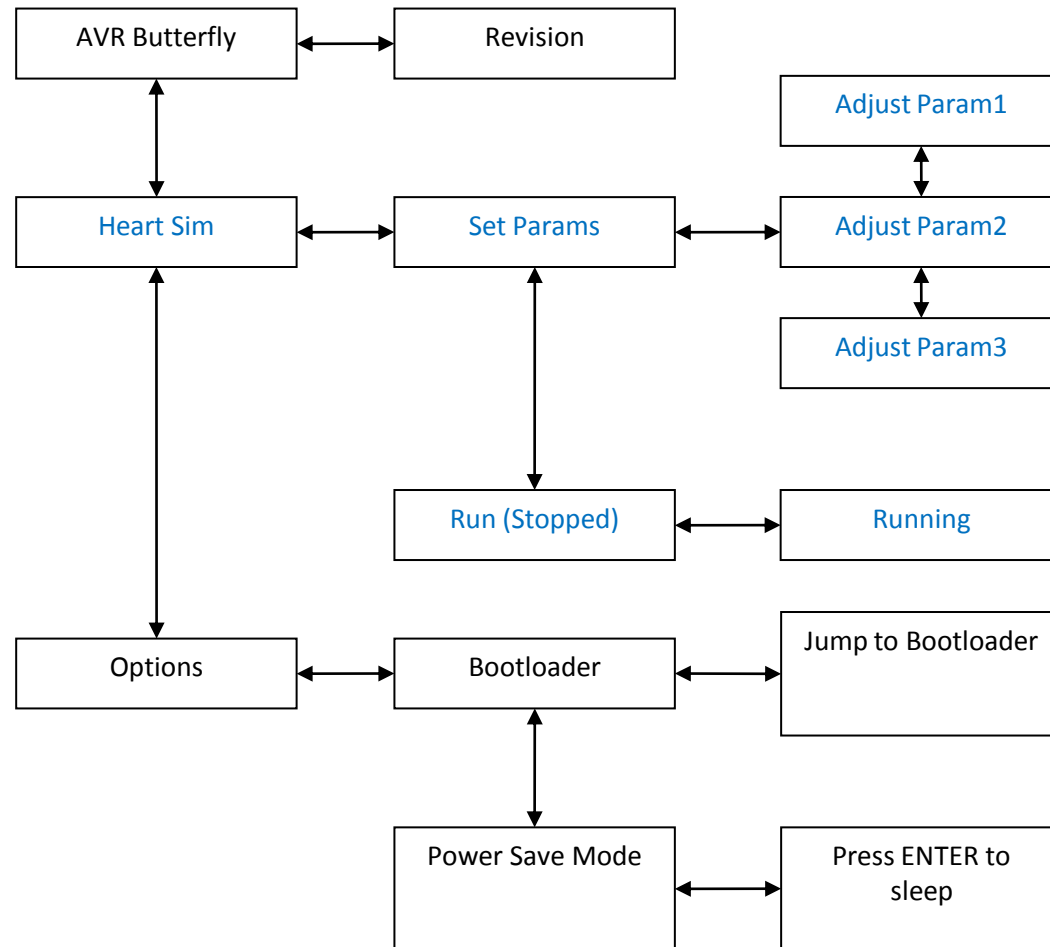
Programming with AVR/Butterfly Features – Clocks

- Two types of Clocks
 - Real-Time Clock
 - From the 32,768 Hz oscillator on board
 - Accurate
 - Work by interrupts
 - CPU busy waiting
 - Functions provided by the GCC-Port
 - The “Delay” function in main.c is an example
 - util/delay.h and util/delay_basics.h from AVR-libc
 - <http://www.nongnu.org/avr-libc/user-manual/modules.html>
- Examples of using clocks is in sound.h
 - Playing music requires timing
 - Register functions to a timer timeout interrupt
 - The functions will be called when timeout happens

Programming with AVR/Butterfly Features – Playing Sounds

- Three component of a sound
 - Frequency, tempo, and duration
- Steps
 - Setting these variables
 - Outputting them to Port OCIA

Programming with AVR/Butterfly Features – A Possible Menu System





RESOURCES

Books and Documents

- AMTEL. (2006, July). *8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash*.
http://www.atmel.com/dyn/resources/prod_documents/doc2514.pdf
- AMTEL. (2005, April). *AVR Butterfly Evaluation Kit User Guide*.
http://www.atmel.com/dyn/resources/prod_documents/doc4271.pdf
- AMTEL. (2003, May). *AVR Butterfly Quick Start User Guide*.
http://www.atmel.com/dyn/resources/prod_documents/doc4249.pdf
- Boehnlein, A. (2003, October 11). *Programming the Butterfly for Idiots Like Me*. http://www.siwawi.arubi.uni-kl.de/avr_projects/pbfi.pdf
- Mckain, W. (2004, May 14). *How to Program the AVR Butterfly*.
<http://www.coe.uncc.edu/~jmconrad/AVR/ButterflyGuide.pdf>
- Pardue, J. (2005). *C Programming for Microcontrollers Featuring ATMEL's AVR Butterfly and the free WinAVR Compiler*. Smiley Micros.

Websites

- AVR Projects website
 - http://www.siwawi.arubi.uni-kl.de/avr_projects
- AVR libc
 - website <http://www.nongnu.org/avr-libc/>
 - reference manual <http://www.nongnu.org/avr-libc/user-manual/index.html>
- WinAVR
 - website <http://winavr.sourceforge.net/>
 - document page <http://winavr.sourceforge.net/helpme.html>
- AVR Freaks website
 - <http://www.avrfreaks.net/>



PROJECT LOGISTICS

Project Logistics

- Milestone 3
 - Due date: April 12th
 - Lab access will be provided
- Milestone 4
 - Using profiling data (as well as arguments from other aspect to establish the correctness result using assurance cases
- Final Presentation
 - May 10th, 9am-11am