Project WEAVER
Wi-Fi Enabled
Active Video
Experimental Rover

Senior Project by:
Amber Blake
Janos Opera
Tyler Lloyd
Project Tasks

Hardware
- Obstacle Detection and Avoidance
- Wireless Interface
- Power Regulation
- Video Interface
- DSP Processor
- Motors and Interfaces
  - H-Bridge Motor Driver
- PCB Design and Verification
- Assembly

Software
- Obstacle Avoidance
- PC Wireless Implementation
- PC Video Stream Viewer
- DSP Video Pre-Processing
- PC Movement Control
- Rover Wireless Implementation
- DSP Movement Control
- GUI Wrapper

Schedule
- Hardware
- Software
Obstacle Detection and Avoidance

Two Components – Hardware and Software

Hardware on Car

- IR LEDs, receivers, and resistors
  IR receivers create a signal when the reflected IR signal is detected
- Input from HPWM on DSP
  Duty cycle controls power level to LEDs which changes the sensitivity
- Output to a general purpose I/O pin
  Receiver signal sent back to DSP
Wireless Communication Link

On the Car

- Hardware

  The Plan: PC card, connectors, and bus to the DSP.

  Back-up: Wireless Flash or chip set

- Reference Design

- Software on the DSP
Power Regulation

Sources
- 9.6V Ni-Cad rechargeable (Motors)
- 9V alkaline 6 AA cells in series (Digital)

Power sequencing
- Digital power is available first
- Motor driver power after H-Bridge is initialized

Single switch disables all voltage sources

Digital voltage regulation
- DSP
- Camera
- Wireless card
- Obstacle avoidance
Video Interface

CCD Camera
- Toshiba TW10794V Digital Camera
- Ribbon Cable Input/Output and Power

MPEG Codec
- NEC µPD61051 MPEG2 Hardware Encoder
  208 pin TQFP (Thin Quad Flat Pack) package
  2.5 and 3.3 V requirements
  Samples en route
- Native Streaming Video Output

Mounting Hardware
- Nuts and Bolts.
Embedded Processor

Digital Signal Processor (DSP)
- Motorola DSP56F807
- On chip interfaces
  - SPI
  - UART
  - CAN
  - SCI
  - JTAG
- External Memory interface port
  - 16bit data 16 bit address
- We have a development kit for the 801 version
- We have samples of the 805 and 807 versions
Motors and Interfaces

One motor for forward and reverse
One motor for left and right
One motor for High and Low gear

Hardware Pulse Width Modulator (HPWM) to apply the signals to move

HPWM duty cycle will determine speed.
- Duty Cycle range 0% - 100%
- Provides maximum motor speed

Rotation is proportional to the current passing through it
H-Bridge Motor Driver

Allows bidirectional operation with minimal control signals

Provide a “break” to stop the rotation of the motor

requires one HPWM and either 2 or 4 IO pins
PCB Design and Verification

Protel 99se schematic and layout environment
  - Output Gerber files

Verification, Review
  - Schematic prior to PCB production
  - Analyze current requirements to size traces
  - Component footprints
  - PCB trace paths

Circuit Graphics PCB fabrication
Assembly

Populate the PCB in phases to check power regulation and control
Complete PCB
Mount modules to the Rover
  – PCB
  – Camera
  – Wireless Card
Power all systems and verify power capacity
Obstacle Detection and Avoidance

Software on DSP

- Sensor Control
  
  Input – Approximate speed from Motor Control Software (duty-cycle of the HPWM)
  
  Output – Duty cycle of the HPWM input to IR LEDs

- Evasive Action
  
  Input – Output of the IR sensors
  
  Output – Communication to Motor Control Software to avoid the object
  
  Obstacles avoided by either turning the car, applying the motor breaks, or both

- Allow this feature to be disabled from the PC
PC Wireless Implementation

802.11g wireless network (54 Mbps)
Hardware – PC card
Software – Write to and read from as a socket
  - Wireless input - Video sent from the Rover
  - Socket output – Relay Video to the PC
  - Socket Input – Command Codes sent from the PC
  - Wireless output – Relay Commands to the Rover
  - UDP transfer protocol

There will also be a wireless access point that will act as a network switch.
PC Video Stream Viewer

Video stream viewer
- Input – MPEG video stream from the car received through Wireless Communication Software
- Output – Streaming video display on the monitor
- Decode using Windows Media Player or similar

Create a viewer in a reserved section of the GUI
DSP Video Interface

Initialize CCD Camera
- Send configuration op-codes
  Frame-rate, Resolution, Etc.

Initialize MPEG encoder
- Define input characteristics (Digital)
- Set compression type (MPEG2)
- Set output to real-time stream

Receive stream into DSP from encoder

Make stream available for wireless transmission
PC Motion Control

Input
- Keyboard
- Explore joystick option
- Variable speeds
  - Forward/Reverse
  - Left/Right

Decode into hardware commands
Send to wireless link
Design a GUI for motion control
Rover Wireless Implementation

Hardware – PC card in a connector mounted to the PCB

Software level 1
- Interface
  IDE
  Card Bus

Software Level 2
- Write to and read from as a socket
  Wireless input - Command Codes sent from the PC
  Socket output – Relay Commands to the Rover
  Socket Input – Video sent from the Rover
  Wireless output – Relay Video to the PC
- UDP transfer protocol
DSP Movement Control

Receive Command message from the PC
Decode message
Respond to the message
  – Change the duty cycle of one or more HPWMs
  – Apply the break
  – Change direction Forward/Reverse
Stop if wireless connection is lost
## Hardware Schedule

<table>
<thead>
<tr>
<th>Tasks</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>August</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>point-person</td>
<td>1</td>
<td>21</td>
<td>28</td>
</tr>
<tr>
<td>Power regulation</td>
<td>Janos</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP circuitry</td>
<td>Tyler</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless transmission</td>
<td>Amber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera interface circuitry</td>
<td>Janos</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collision avoidance circuitry</td>
<td>Amber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor driver</td>
<td>Tyler</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCB design</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCB build</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part acquisition</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic assembly</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Tasks and point-persons:*
- Power regulation: Janos
- DSP circuitry: Tyler
- Wireless transmission circuitry: Amber
- Camera interface circuitry: Janos
- Collision avoidance circuitry: Amber
- Motor driver: Tyler
- PCB design and PCB build: All
- Part acquisition and Basic assembly: All
## Software Schedule

<table>
<thead>
<tr>
<th>Project</th>
<th>September 1</th>
<th>September 2</th>
<th>October 1</th>
<th>October 2</th>
<th>October 3</th>
<th>October 4</th>
<th>October 5</th>
<th>November 1</th>
<th>November 2</th>
<th>November 3</th>
<th>November 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP Wireless communication</td>
<td>Tyler</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>DSP Motor control</td>
<td>Tyler</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>DSP Video interface</td>
<td>Janos</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>DSP Sensor processing and collision avoidance</td>
<td>Amber</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>SP MPEG encoding PC laptop Wireless communication</td>
<td>Amber</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>PC laptop Motor control</td>
<td>Janos</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>PC laptop Video interface</td>
<td>Amber</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>PC laptop GUI wrapper</td>
<td>All</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
<td>Orange</td>
<td>Red</td>
<td>Red</td>
<td>Orange</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>point-person</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Colors indicate progress status.