The Car Jackers

By Jeremy Bonnell
Tong Wu
Introduction

Remote controlled car functions:

- Heating and Cooling
- Start and Stop engine
- Door Lock and Unlock
General Layout
Arduino Mega 2560

**Power:** 5V (Recommended 7 - 12 V)

3.3V supply generated

**Memory:** 256KB Flash

**I/O:** 54 Digital (4 UARTs) at 5 V.

**IDE:** Arduino 1.0
Cellular Shield SM5100B

**Power**: 3.3V - 4.2V

**I/O**: UARTs up to 460kbps

Operates at 1900 MHz
- Quad-band Wired Cellular Antenna SMA
Antenna

- Frequency: 1900MHz
- Gain: 3.5 dBi
On Board Diagnostic System (OBD-II)

**Project Functionality**
- To check engine of car

**Output:** Serial port (RS-232)
Connections

- Same power, ground, reset, etc.

- Use Serial ports

- Digital pins on the side (pins 51-53 SPI for digital potentiometer)
Connections (diagrams)
Software Design

Android App. - GUI sends the messages to the cellular shield. Also receives the status.

Arduino - Parses messages from Android phone then performs tasks. Also sends the status from OBD-II to the phone.
Software: Arduino

- Send (AT commands) to Serial 1
  ○ Set frequency to 9600 baud
  ○ Set text mode, send SMS with the number and message

- Receive values from Serial 1
  ○ Set output
  ○ Parse message
Android Application

- SMS send
- SMS receive then parse or display
- Onclick handlers for sending SMS with the buttons
## Messages and Tasks

<table>
<thead>
<tr>
<th>Button Name / Task</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Engine</td>
<td>a0</td>
</tr>
<tr>
<td>Stop Engine</td>
<td>a1</td>
</tr>
<tr>
<td>Lock Door</td>
<td>a2</td>
</tr>
<tr>
<td>Unlock Door</td>
<td>a3</td>
</tr>
<tr>
<td>Defrost</td>
<td>b0</td>
</tr>
<tr>
<td>Temp Up</td>
<td>a4</td>
</tr>
<tr>
<td>Temp Down</td>
<td>a5</td>
</tr>
<tr>
<td>Fan Up</td>
<td>a6</td>
</tr>
<tr>
<td>Fan Down</td>
<td>a7</td>
</tr>
<tr>
<td>Override On</td>
<td>a8</td>
</tr>
<tr>
<td>Override Off</td>
<td>a9</td>
</tr>
</tbody>
</table>
Hardware Design

Diagram of Hardware Design:
- **Android Phone**
  - **SIM CARD**
  - **Cellular Shield**
    - **Arduino MCU**
      - **DigitalWrite** (0V-3V, 50mA)
      - **AnalogRead** (0V-5V, 2A)
    - **Relay PCB**
      - **Opto-Isolators**
    - **Components**
      - **Starter**
      - **Fan**
      - **Temp Vent**
      - **R-Defroster**
      - **Vents**
      - **Locks**
    - **OBDII Interface**
      - **Ignition**
      - **Fan Control**
      - **Temp Control**
      - **Defrost Control**
      - **Vent Control**
      - **Door Locks**
    - **Disable Mobile Control**
    - **Vehicle Controls**

**Power Supply**: +12V
Relay Driver

- PCB design

- Amplify Signals from Arduino

- Override signal by Arduino used as SELECT line for SPDT Relay Switches

- Operates like multiplexer and selects between dashboard controls or Arduino logic
Switch Relay

Type-A Logic (Amplifier Circuit)

- Digital Pin +5V
- +12V

SQ2310ES Mosfet
- 10KΩ
- 275-226 30A Automotive Relay Bank
- Opto-Isolator
- Starter Motor

+12V

Ignition Switch

Open when ignition not turned
Amplifier/Switch Relay

Type-B Logic (Switch/Amplifier Circuit)
AD5206 Digital Potentiometer

- 256 positions
- Multiple independently programmable channels
- 6-channel
- Terminal resistance of 10 kΩ, 50 kΩ, or 100 kΩ
- 3-wire SPI-compatible serial data input
- +2.7 V to +5.5 V single-supply operation
Reverse Engineering

- Measure actual voltage and resistor values from each setting of dashboard controls. Typical +12V
  - Digital potentiometer values stored in Arduino software for each setting. Used as voltage divider from 5V digital output then amplified by Relay PCB to proper levels

- Translate OBDII codes (Ignition status to Arduino)

- Door locks (Monitor) - work in tandem with actual locks.

- Locate seat sensor - All Mobile controls disabled after person in driver's seat. 5 second timeout on Ignition Control
2005 Toyota Camry Wiring Diagram - B
Task List And Schedule

1. Communication between Phone and Arduino      Complete

By July 31, 2012:
2. Reverse Engineering of Vehicle               2 weeks
3. Build PCB Design for Relay Driver            3 weeks
4. Android Application and Arduino code         3 weeks

By Demo Day:
5. Connect & Test: Arduino and Relay Driver Interface 4 weeks
6. Connect & Test: Car and Relay Driver Interface 6 weeks
7. System Testing: (Phone to Car)               4 weeks
8. Finishing touches: System testing and UI upgrade 2 weeks
# Components List (B.O.M.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toyota Camry</td>
<td>$...</td>
</tr>
<tr>
<td>Android Phone</td>
<td>$...</td>
</tr>
<tr>
<td>Sim Card (3) - AT&amp;T</td>
<td>$  75</td>
</tr>
<tr>
<td>Arduino Mega 2560 - Sparkfun</td>
<td>$  70</td>
</tr>
<tr>
<td>Cellular Shield SM5100B - Sparkfun</td>
<td>$100</td>
</tr>
<tr>
<td>PCB (relay driver) - Altium</td>
<td>$100</td>
</tr>
<tr>
<td>OBD-II - uResearch</td>
<td>$  20</td>
</tr>
<tr>
<td>wiring diagrams (alldata.com)</td>
<td>$  27</td>
</tr>
<tr>
<td>Misc. (cables, antenna, etc.)</td>
<td>$  20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$412</strong></td>
</tr>
</tbody>
</table>
Risk Management

- Relay Driver will require a lot of testing before installing. May need to buy blower motor, vent servo motors, etc... to test in lab first

- All PCB components may not be accessible in PCB design library. May have to design own footprints or build relay driver on proto-board instead

- Must make sure all sectors are isolated so back electromagnetic flux does not fry components

- Danger of making car inoperable. Always make wiring diagrams before disassembling
Demo Day

- Vehicle parked outside in parking lot
- Need about 10 mins to show all the functions
- Android interface sends SMS to Cellular Shield
-Parsed to Arduino
- Arduino sends signal to Relay Driver
- Stuff happens
Questions