

# Remote Vehicle Interface (RVI)

Travis Johnson

Ben Moon

# Project Goals

## **Interface with vehicle using phone or WWW**

- **Start / Kill engine**
- **Lock / Unlock doors**
- **Pop trunk**

**Must provide reasonable security**

**Mimic satellite-based implementation**

- **Top-down LOS channel model**

# Overall Design

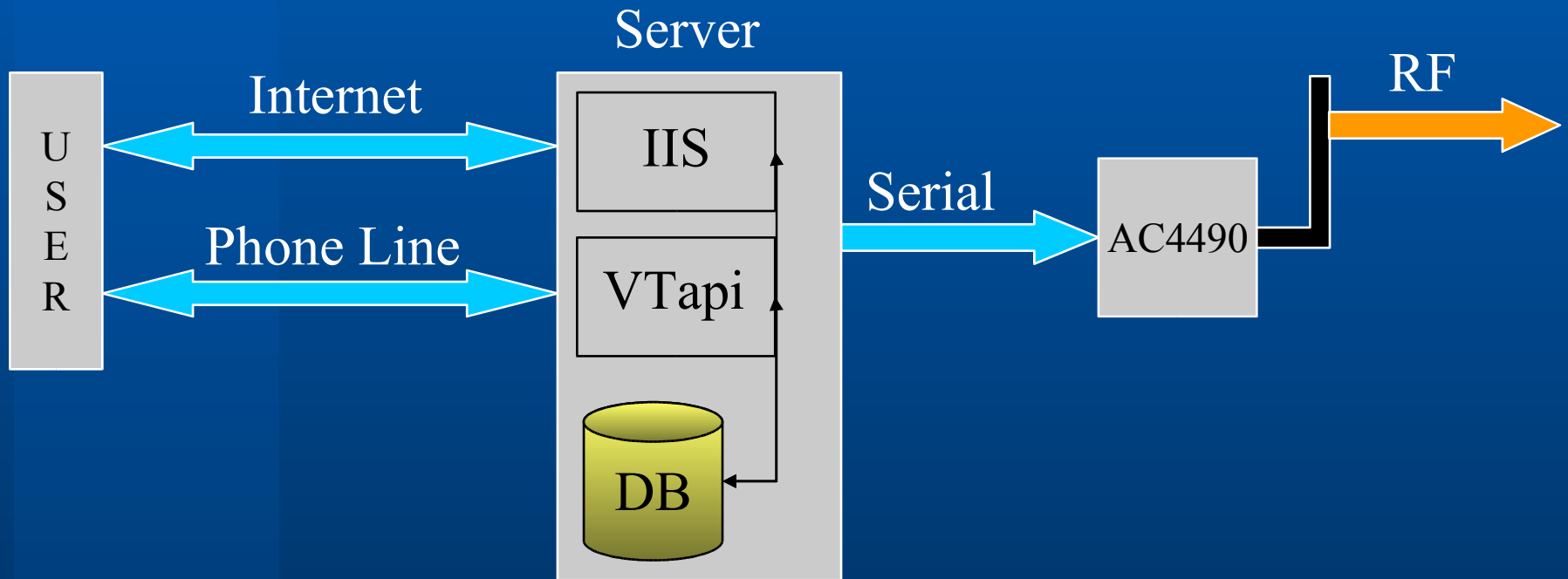
## RVI Server (TX)

- Stores user database
- Provides user interfaces
- Authenticates users
- Transmits RF commands

## Vehicle Control Unit (RX)

- Accepts RF commands
- Interfaces with vehicle systems

# RVI Server – Block Diagram



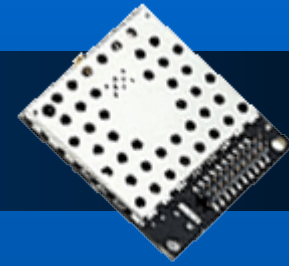
# RVI Server – Details 1



## PC Server

- MS Access for user database
- VTapi ActiveX control for Dial-in
  - Custom message prompts (.wav)
  - Tone detection
  - Thread-safe
- Microsoft IIS for Web UI hosting
  - Dynamic content with ASP.NET
  - Connection to database via **ADO.NET**

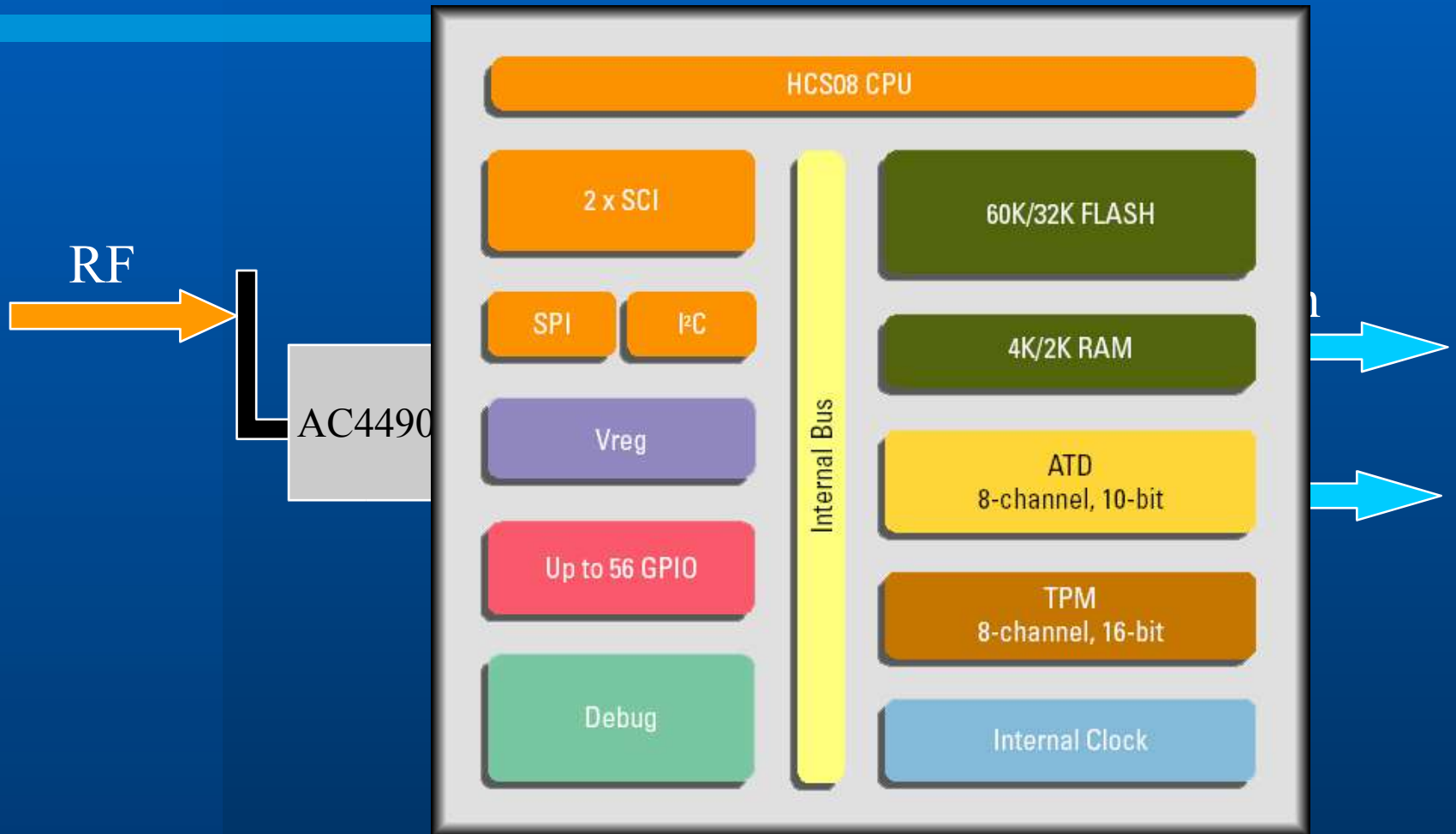
# RVI Server – Details 2



## **TX - AC4490 Transceiver**

- **Frequency: 902 - 928 MHz**
  - License-free band
- **Power: 5 - 1000 mW (variable)**
- **Range: ~20 miles (LOS)**
  - at 500 mW
  - with 3 dB omni antenna
- **Modulation: FHSS FSK**
- **RF Data Rate: 76.8 kbps**
- **Supply Voltage: 3.3 Vdc**

# Vehicle Control Unit



# Vehicle Control Unit – Details 1

## HCS08-GB60 MCU

- 8-bit microprocessor
- Operates down to 1.8 V
- Real-time debugging
  - No additional hardware
  - Uses one SCI port
- CodeWarrior IDE included
  - with C compiler !!





# Vehicle Control Unit – Details 2

## **RX - AC4490 Transceiver**

- Same specs as RVI-server TX

# Specific Parts 1

Accessories: AC1400 TX/RX

The screenshot displays a debugger interface with several panels:

- Source:** Shows C code for `_Startup()` in `Start08.c`. The code includes comments and a `#ifndef` block for the `ELF OBJECT FILE FORMAT`. A line of code is highlighted: `DisableInterrupts: /* in HIWARE format, this is done in the prestart code */`.
- Assembly:** Shows the assembly code for the `_Startup` function, starting with `18A7 SEI` and including instructions like `LDA`, `BIT`, `ENE`, `LDEH`, `TXS`, `B3R`, `LDEH`, and `J3R`.
- Register:** Shows the state of registers for the `HCS08` processor. The `PC` register is highlighted at `18A7`. Other registers shown include `A`, `HX`, `SR`, and `SP`.
- Data1:** Shows data symbols for `_startupData` and `_SEG_END_5STACK`.
- Data2:** Shows data symbols for `_Startup`.
- Memory:** Shows a memory dump with addresses and hex values, such as `0000 00 0A FF DE 00 00 7F 25`.
- Command:** Shows the command `!savebp off`.



# Specific Parts 2

## PC Server

- PIII-500 CPU
- Abit motherboard
- 256 MB PC100 RAM
- 3Com 3c905b NIC
- Diamond SupraExpress 56i voice modem
- a.k.a. “Travis’s old desktop”



# Security Overview

## Security

- Each user has a unique seed and an offset for a random # algorithm
  - MCU and DB retain these numbers
- MCU checks the authorization # received against the next 128 possibles in the series (using last successful offset)
  - Why so many?
- If successful, it resynchronizes

# Issues 1

## Vehicle installation

- Will get Best Buy, Circuit City, etc. to “donate” a tutorial over the summer

## Signal strength

- Range is everything
- Panel antennas (**example**)
  - Modest gain
  - Can be mounted discretely on exterior
  - Prevents losses caused by metal exterior

# Issues 2

## Serial Port Communication

- Never written code that uses serial port to transmit data
- Obviously, both UI's need a way to do this
- Shouldn't be too hard
  - Famous last words...

# Schedule

Task	August	September	October	November	December
User DB	█				
Phone UI	█				
Web UI		█			
Server <===> TX	█	█			
Program MCU			█		
RX <===> MCU			█		
MCU <===> Vehicle				█	
Testing & Tweaking					█
Documentation	█	█	█	█	█

# Milestones 1

## 1. Access database built

- User ID#
- Password
- Seed for random number generator

## 2. Working Dial-in UI

- Accept calls
- Authenticate users
- All menus functional
- Packets getting to serial port



# Milestones 2

## 1. Working Web UI

- Users can log in
- All menus functional
- Packets getting to serial port

## 2. Server and TX integrated

- TX receiving packets on serial port
- Packets being transmitted correctly

# Milestones 3

## 1. MCU Programmed

- Receiving packets on SCI port
- Checking authorization #
- Setting GPIO pins properly

## 2. RX and MCU integrated

- RX receiving RF packets
- Forwarding to serial port
- MCU still receiving

# Milestones 4

## 1. MCU and Vehicle integrated

- Able to start/kill engine
- Able to lock/unlock doors
- Able to pop trunk
- All work at LONG range
  - Say 1 mile for starters

# That's it...

---

**Questions?**

**Suggestions...**