

```
#include <fcntl.h>
#include <sys/termios.h>
#i
#i
#i
Se
(
if(serialPort <= 0)
{
    close(serialPort);
    throw std::runtime_error("serial port could not be opened.");
}

struct
// get
tcgetat
// set
cfsetispeed(&terminalSettings, (speed_t)baudRate);
cfsetospeed(&terminalSettings, (speed_t)baudRate);

terminalSettings.c_cflag |= CREAD & ~CLOCAL;

// SN1
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag |= C56; // set for 8 bits.
```

University of West Florida School of Science and Engineering Unmanned Systems Group



University of
West Florida

Unmanned Systems Group

Vision Statement

- With respect to **Unmanned Systems**, the vision of the University of West Florida is Clear:

To become the best regional comprehensive university in America

- This requires a commitment by those who deeply believe in **UWF**, its success, and to understand that investments today will create great futures for our students, university, and region tomorrow.



Unmanned Systems

An Innovative New Beginning for the Schools of Science and Engineering

In response to the escalating need for graduates who can meet rapidly changing industry demands in scientific fields critical to building and maintaining a strong regional and national economy, the University of West Florida has established a new School of Science and Engineering



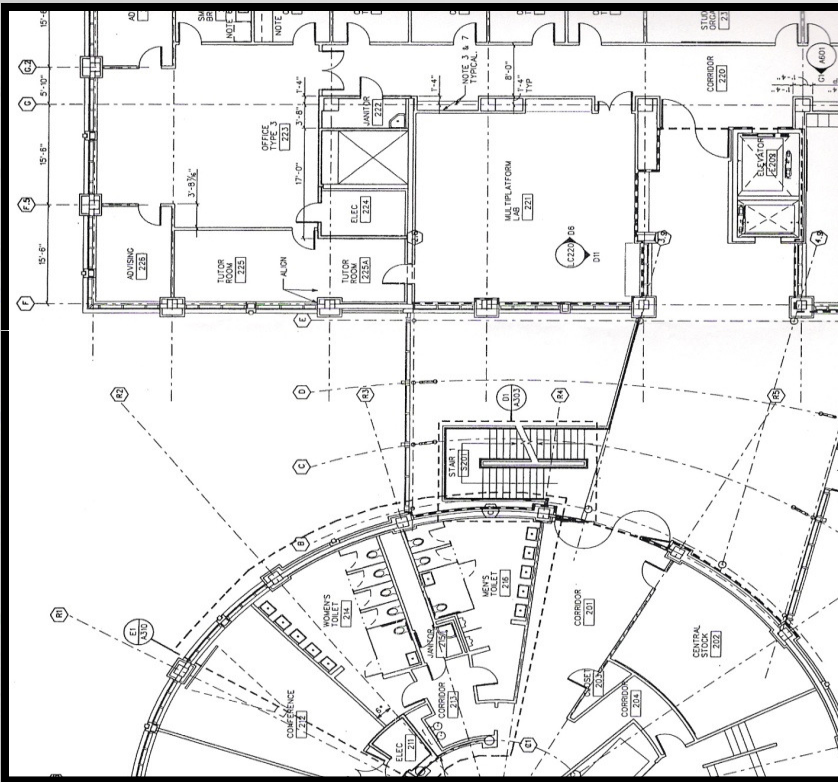
University of West Florida

School of Science and Engineering



University of
West Florida

School of Science and Engineering Unmanned Systems Lab



- Maximum visibility on 1st floor of 4-story atrium behind glass enclosure
- Focal point of both North and South entrance ways, elevator, and stairway
- Averaging 1 new applicant per day
- Equipped with oscilloscopes, logic analyzers, DVM's and other test equipment not unlike industry

```
#include <fcntl.h>
#ir
#ir
#ir
#ir
```

University of West Florida

School of Science and Engineering

```
SerialInterface::SerialInterface(std::string device, int baudRate)
{
    serialPort = open(device.c_str(), O_RDWR | O_NOCTTY | O_SYNC);
    if (serialPort == -1)
    {
        close(serialPort);
        throw std::runtime_error("Unable to open serial port");
    }
    struct termios terminalSettings;
    // get the current settings
    tcgetattr(serialPort, &terminalSettings);
    // ...
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);
    terminalSettings.c_cflag |= CREAD & ~CLOCAL;
    // SN1
    terminalSettings.c_cflag &= ~...
    terminalSettings.c_cflag &= ~...
    terminalSettings.c_cflag &= ~...
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```

Successful 21st century science, technology, engineering, and math (STEM) programs **must** focus attention on learning environments in which all students have ample opportunity for practicing science and sharing their discoveries.

UWF's New School of Science and Engineering is answering that need!



```
#include <fcntl.h>
#ir
#ir
#ir
#ir
```

UWF's School of Science and Engineering and PKAL

```
SerialInterface::SerialInterface(std::string device, int baudRate)
{
    serialPort = open(device.c_str(), O_RDWR | O_NOCTTY);
    if (serialPort == -1)
    {
        close(serialPort);
        throw std::runtime_error("Serial port could not be opened.");
    }
    struct termios terminalSettings;
    // get the current settings
    tcgetattr(serialPort, &terminalSettings);
    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CREAD & ~CLOCAL;

    // 8N1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```

The philosophy and concepts guiding UWF's new School of Science and Engineering are based on Project Kaleidoscope (PKAL), the national initiative funded by the National Science Foundation and the Keck Foundation. Their mission and activities support and enhance collaborative and interdisciplinary education for those disciplines of STEM.



UWF Unmanned Systems Laboratory

The UWF Unmanned Systems Laboratory is the proud home of an award-winning Autonomous Water Vehicle team and Unmanned Aerial Systems team as well as the Unmanned Ground Systems and Tour Robot Project for the new School of Science and Engineering. This school is active in engineering research in partnership with local business, other universities, and government agencies.



Unmanned Systems—Goals

- Inspire students to achieve their full potential through an engineering experience.
- To develop research, development, and management skills to enhance for greater industry acceptance.
- Greater retention of pre-engineering students.
- Achieve purposeful and sustainable growth.



Method to Accomplish the Mission through Engineering Projects

- Unmanned Systems Projects must:
 - Require research to accomplish the mission
 - Require development of systems to accomplish the mission
 - Develop student management skills
 - Require a teams of students and teamwork
 - Have a deadline
 - Have budget limitations
- Success of Project produces team awards and increase status of the overall program



```
#include <fcntl.h>
#include
#include
#include
#include
```

<http://robotics.ece.uwf.edu/home.php#>

```
SerialInterface::SerialInterface(std::string device, int baudRate)
```

```
{
    serialPort = open(
```

```
if(serialPort <= 0
{
    close(serialPort
    throw std::runti
}
```

```
struct termios ter
```

```
// get the current
tcgetattr(serialPo
```

```
// set baud rate
cfsetispeed(&termi
cfsetospeed(&termi
```

```
terminalSettings.c_cflag |= CREAD & ~CLOCAL;
```

```
// SN1
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= C56; // set for 8 bits.
```



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"
```

```
SerialInterface::SerialInterface(std::string device, int baudRate)
```

```
{
    serialPort = open(device.c_str(), O_RDWR | O_NOCTTY | O_NDELAY);
```

Autonomous Unmanned Vehicles Systems International (AUVSI)

Unmanned Aerial Systems (UAS)

AUVSI Competition 2009

```
struct termios terminalSettings;

// get the current settings
tcgetattr(serialPort, &terminalSettings);

// set baud rate
cfsetispeed(&terminalSettings, (speed_t)baudRate);
cfsetospeed(&terminalSettings, (speed_t)baudRate);

terminalSettings.c_cflag |= CREAD & ~CLOCAL;

// SN1
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



University of
West Florida

Unmanned Aerial Systems (UAS)

- Competed and placed 2nd in Flight Competition out of 25 Colleges and Universities in the Association for Unmanned Vehicle Systems International (AUVSI)—June 2009
- AUVSI is the world's largest non-profit organization devoted exclusively to advancing the unmanned systems
- Received \$8000.00 in prize money
- Placed among large well-known universities
- Completed more bonus tasks than any other university in the history of the competition
- Only aircraft that took off, performed the mission and landed without any action from the safety operator
- Primary objectives of the competition were for each team to build an unmanned aircraft to fly autonomously, navigate a specified course and use onboard sensors to locate and assess a series of man made objects on the ground prior to returning to the launch point for landing.



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc
```

```
#include "Serial
```

```
SerialInterface:
```

```
{
```

```
serialPort = c
```

```
if(serialPort
```

```
{
    close(serial
    throw std::r
}
```

```
struct termios
```

```
// get the cur
tcgetattr(seri
```

```
// set baud rate
```

```
cfsetispeed(&terminalSettings, (speed_t)baudRate);
```

```
cfsetospeed(&terminalSettings, (speed_t)baudRate);
```

```
terminalSettings_
```

```
// SN1
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



UWF's 2009 UAS Team

(Left to Right, Clint Edmonson, Niel Edmunson, Eric Becker, Steve Long, David Algeo)



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyS0", "r");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CREX;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

Cornell University



**University of
West Florida**


```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc...

#include "Serial...

SerialInterface:
{
    serialPort = c...

    if(serialPort
    {
        close(serial...
        throw std::r...
    }

    struct termios...

    // get the cur...
    tcgetattr(seri...

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cf...

    // SN1
    terminalSettings.c_cflag &= ...
    terminalSettings.c_cflag &= ...
    terminalSettings.c_cflag &= ...
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

Embry-Riddle Aeronautical University



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyS0", "r+");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CLOCAL;
    terminalSettings.c_cflag |= CREAD;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag &= ~CSIZE;
    terminalSettings.c_cflag |= CS8;
    // set for 8 bits.
```



AUVSI 2009

University of New Delhi, India



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = open("/dev/ttyS0", O_RDWR | O_NOCTTY);

    if(serialPort < 0)
    {
        close(serialPort);
        throw std::runtime_error("Failed to open serial port");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CLOCAL;

    // SN1
    terminalSettings.c_cflag |= CRTSCTS;
    terminalSettings.c_cflag |= CS8;
    // set for 8 bits.
```



AUVSI 2009

Johns Hopkins University



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyUSB0", "r");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CREAD;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

Mississippi State



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyUSB0", "r");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CRTSCTS;

    // SN1
    terminalSettings.c_cflag &= ~CRTSCTS;
    terminalSettings.c_cflag &= ~CRTSCTS;
    terminalSettings.c_cflag &= ~CRTSCTS;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

North Carolina State



University of
West Florida

```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyUSB0", "r");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

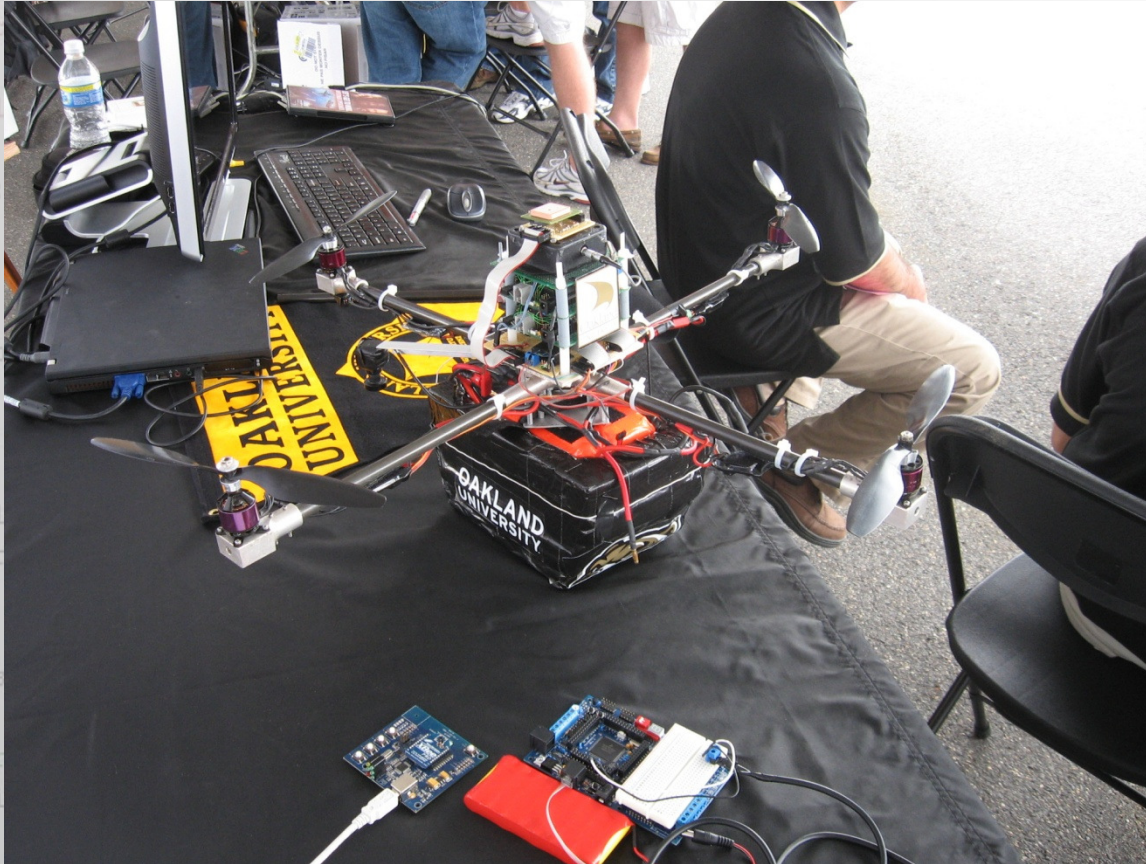
    struct termios terminalSettings;
    tcgetattr(serialPort, &terminalSettings);

    // get the current baud rate
    tcgetattr(serialPort, &terminalSettings, &baudRate);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CRTSCTS;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

Oakland University



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = 0;

    if(serialPort < 0)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CLOCAL;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

University of California at San Diego



University of
West Florida

```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyUSB0", "r");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;
    tcgetattr(serialPort, &terminalSettings);

    // get the current baud rate
    tcgetattr(serialPort, &terminalSettings, &baudRate);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CREX;

    // SN1
    terminalSettings.c_cflag &= ~CS8;
    terminalSettings.c_cflag &= CS6;
    terminalSettings.c_cflag &= CRTSCTS;
    terminalSettings.c_cflag |= CSD;
    tcsetattr(serialPort, TCSANOW, &terminalSettings);

    // set for 8 bits.

```



AUVSI 2009

University of Utah



**University of
West Florida**


```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface
{
    serialPort = ...

    if(serialPort
    {
        close(serialPort);
        throw std::runtime_error("Serial port closed");
    }

    struct termios
    {
        // get the current settings
        tcgetattr(serialPort, &termios);

        // set baud rate
        cfsetispeed(&terminalSettings, (speed_t)baudRate);
        cfsetospeed(&terminalSettings, (speed_t)baudRate);

        terminalSettings.c_cflag |= ...

        // SN1
        terminalSettings.c_cflag &= ...
        terminalSettings.c_cflag &= ...
        terminalSettings.c_cflag &= ...
        terminalSettings.c_cflag |= CS8; // set for 8 bits.
    }
};
```



AUVSI 2009

Virginia Commonwealth University



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = fopen("/dev/ttyS0", "r+");

    if(serialPort == NULL)
    {
        close(serialPort);
        throw std::runtime_error("Serial port not found");
    }

    struct termios terminalSettings;

    // get the current terminal settings
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |= CRTSCTS;

    // SN1
    terminalSettings.c_cflag &= ~PARENB;
    terminalSettings.c_cflag &= ~PARODD;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



AUVSI 2009

UWF Control Station



University of
West Florida

```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc

#include "Serial

SerialInterface:
{
    serialPort = c
    if(serialPort
    {
        close(serial
        throw std::r
    }

    struct termios
    // get the cur
    tcgetattr(seri

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudRate);
    cfsetospeed(&terminalSettings, (speed_t)baudRate);

    terminalSettings.c_cflag |=
    // SN1
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag |= C56; // set for 8 bits.
```



AUVSI 2009

UWF Ready to Take Flight



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc

#include "Serial

SerialInterface:
{
    serialPort = o
    if(serialPort
    {
        close(serial
        throw std::r
    }
    struct termios
    // get the cur
    tcgetattr(seri

    // set baud rate
    cfsetispeed(&terminalSettings, (speed t)baudRate);
    cfsetospeed(&terminalSettings, (speed t)baudRate);

    terminalSettings

    // SN1
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag &=
    terminalSettings.c_cflag |= C56; // set for 8 bits.
```



AUVSI 2009

UWF Award Ceremony—2nd in Flight—4th Overall



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc
```

```
#include "Serial
```

```
SerialInterface:
```

```
{
```

```
serialPort = c
```

```
if(serialPort
```

```
{
    close(serial
    throw std::r
}
```

```
struct termios
```

```
// get the cur
tcgetattr(seri
```

```
// set baud rate
```

```
cfsetispeed(&terminalSettings, (speed_t)baudRate);
```

```
cfsetospeed(&terminalSettings, (speed_t)baudRate);
```

```
terminalSettings.c_cfl
```

```
// SN1
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= C58; // set for 8 bits.
```



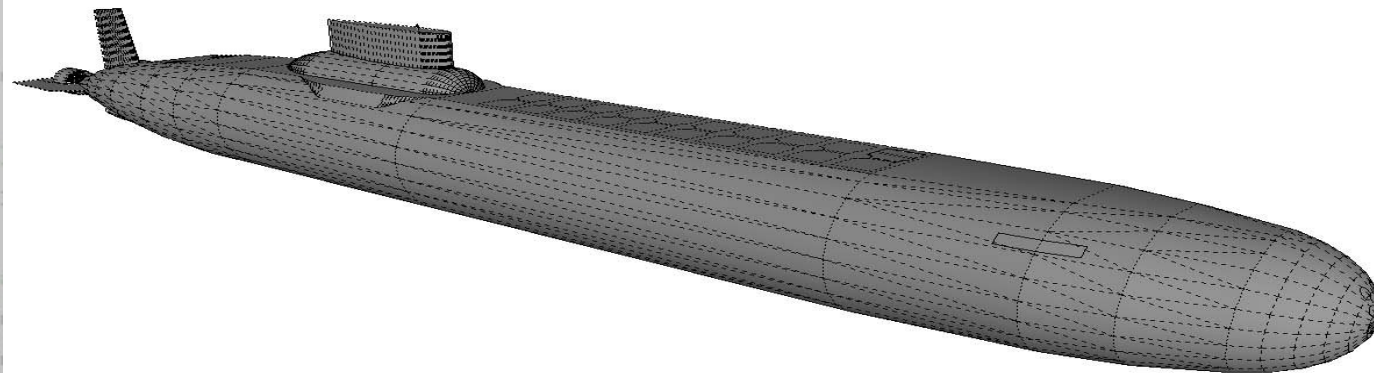
AUVSI 2009

UWF Team and the "Whole Package"



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#inc
#inc
Serial
(
se
if
(
close(serialPort);
throw std::runtime_error("serial port could not be opened.");
)
struct t
// get t
tcgetattr
// set b
cfsetispe
cfsetosp
terminal
// SN1
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag |= C56; // set for 8 bits.
```

University of West Florida School of Science and Engineering Autonomous Underwater Vehicle (AUV)



University of
West Florida

AUVSI Goals

The goals of the AUVSI student competitions are to provide opportunities for students to experience the challenges of system engineering, to develop skill in accomplishing realistic missions with autonomous vehicles and to foster relationships between young engineers and the organizations developing and producing autonomous vehicle technologies.



AUVSI Student Competitions

The primary emphases of the AUVSI student competitions are learning and outreach. These events are not grand challenges designed explicitly to progress the state-of-the-art. The objective is to produce the people who will push the envelope in the future. Major innovations may be spawned in these events, but this is a by-product, not an objective. Most important are gaining an appreciation for the trade offs inherent in any system design and the lessons learned in transitioning from a working bench prototype to operating reliably in the real world.



Experience in AUVSI Student Competitions

The legacy of the student competitions can be found today throughout government and industry. Employers and venture capitalists seek out prospects with the kind of resourcefulness and team management experience that former competitors offer.



International Autonomous Underwater Vehicle Competition

Hosted by AUVSI and the Space and Naval Warfare Systems Center
SSC SD TRANSDEC Facility
San Diego, CA



AUV 2009 “Mission”

```
#include <fcntl.h>
#ir
#ir
#ir
#ir
SerialInterface::SerialInterface(std::string device, int baudRate)
{
    Your mission, should you choose to accept it, is the following: Land on
    the beach and fire off a flare for the forces still in the water (to blind
    the enemy's night vision), and proceed under the barb wire. From
    there you have one of two choice: Head towards the targets for the
    bombing run, or head towards the machine gun nests. For the
    bombing run, you'll have the possibility to mark a primary and
    secondary target, or if you can't locate those, there are also two
    targets of opportunity. For the machine gun nest, you'll be required
    to lob up to two grenades into a small square opening. Finally, an
    operative has marked the building with the correct briefcase which
    contains the secret documents with a pinger. You are to follow the
    pinger, retrieve the briefcase, and head to the rooftop for
    extraction.
}
// get
tcgetattr(serialPort, &terminalSettings);
// se
cfsetispeed(&terminalSettings, (speed_t)baudRate);
cfset
```



University of
West Florida

```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdext>

#include "SerialInterface.h"

SerialInterface
{
    serialPort = c

    if(serialPort
    {
        close(serial
        throw std::r
    }

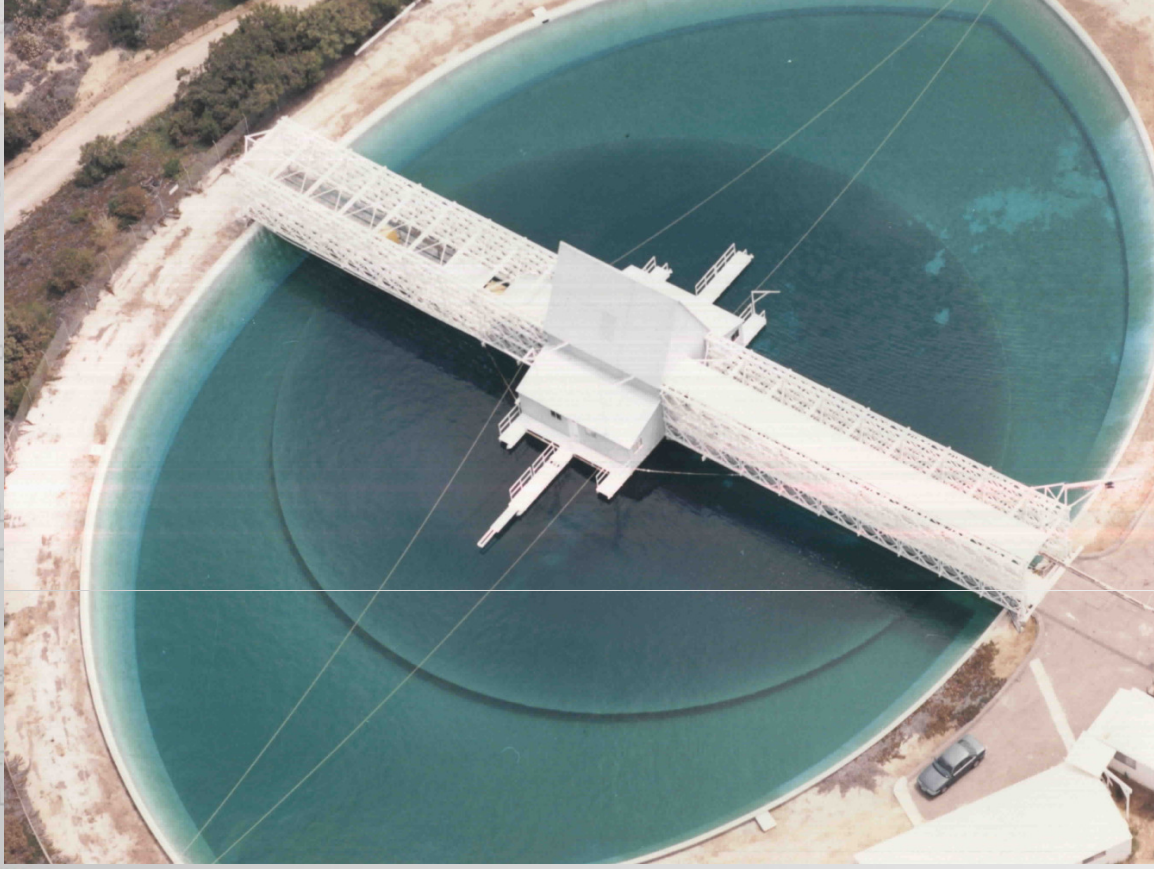
    struct termios

    // get the cur
    tcgetattr(seri

    // set baud rate
    cfsetispeed(&st
    cfsetospeed(&st

    terminalSetting

    // SN1
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag |= CS8; // set for 8 bits.
```



Aerial Photo of SPAWAR Facility

The bridge structure has no piers or supports in the pond.



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc...

#include "Serial

SerialInterface:
{
    serialPort = c

    if(serialPort
    {
        close(serial
        throw std::r
    }

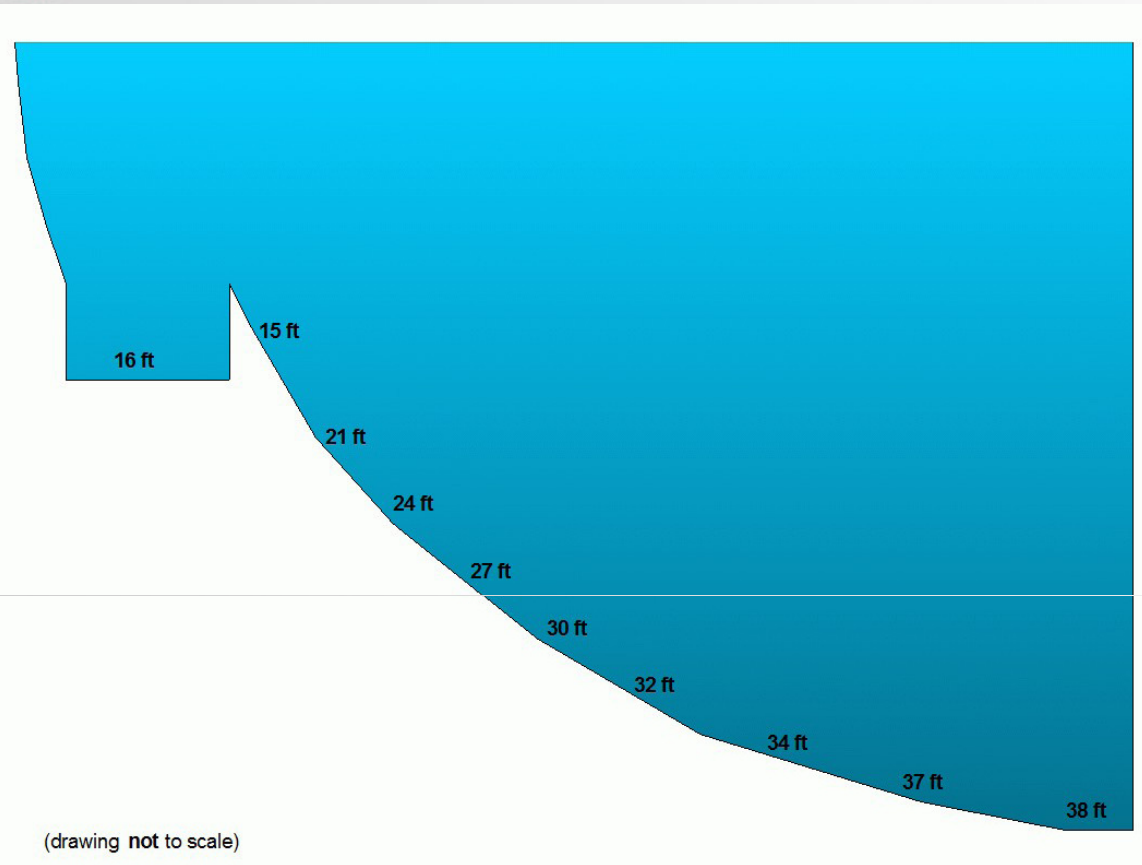
    struct termios

    // get the cur
    tcgetattr(seri

    // set baud rate
    cfsetispeed(&st
    cfsetospeed(&st

    terminalSettings.c_iflag |= C56; // set for 8 bits.

    // SN1
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag &= ~
    terminalSettings.c_cflag |= C56; // set for 8 bits.
```



Cross Section of Arena

The "acoustic trap" varies in width around the pond.



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexec>
```

```
#include "Serial
```

```
SerialInterface:
```

```
(
```

```
serialPort = c
```

```
if(serialPort
```

```
{
```

```
close(serial
```

```
throw std::r
```

```
);
```

```
struct termios
```

```
// get the cur
```

```
tcgetattr(seri
```

```
// set baud rate
```

```
cfsetispeed(&st
```

```
cfsetospeed(&st
```

```
terminalSettings
```

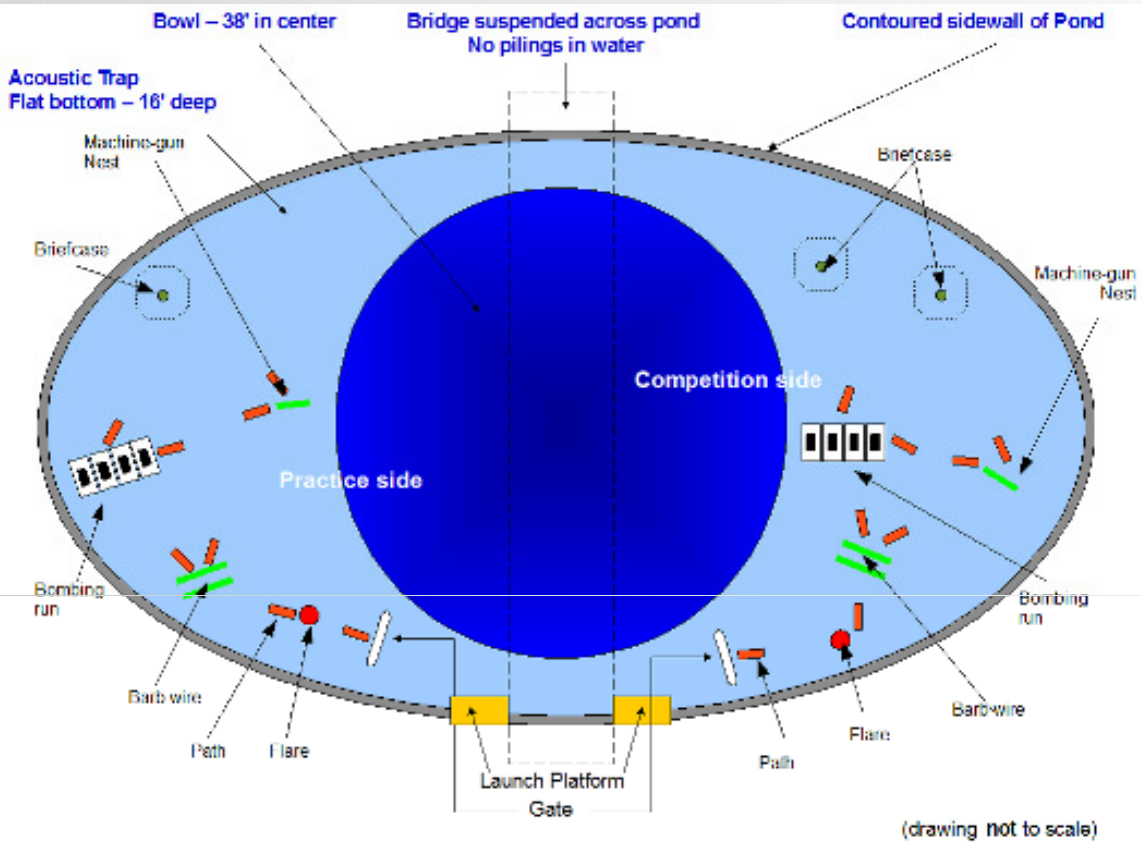
```
// SN1
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= C56; // set for 8 bits.
```



General Layout of Arena

The arena is split into a competition side and a practice side.



```

#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc

```

```

#include "Serial

```

```

SerialInterface:
(

```

```

(

```

```

serialPort = c

```

```

if(serialPort
(

```

```

close(serial
throw std::r
)

```

```

)

```

```

struct termios

```

```

// get the cur
tcgetattr(seri

```

```

// set baud rate
cfsetispeed(&terminalSettings, (speed_t)baudRate);
cfsetospeed(&terminalSettings, (speed_t)baudRate);

```

```

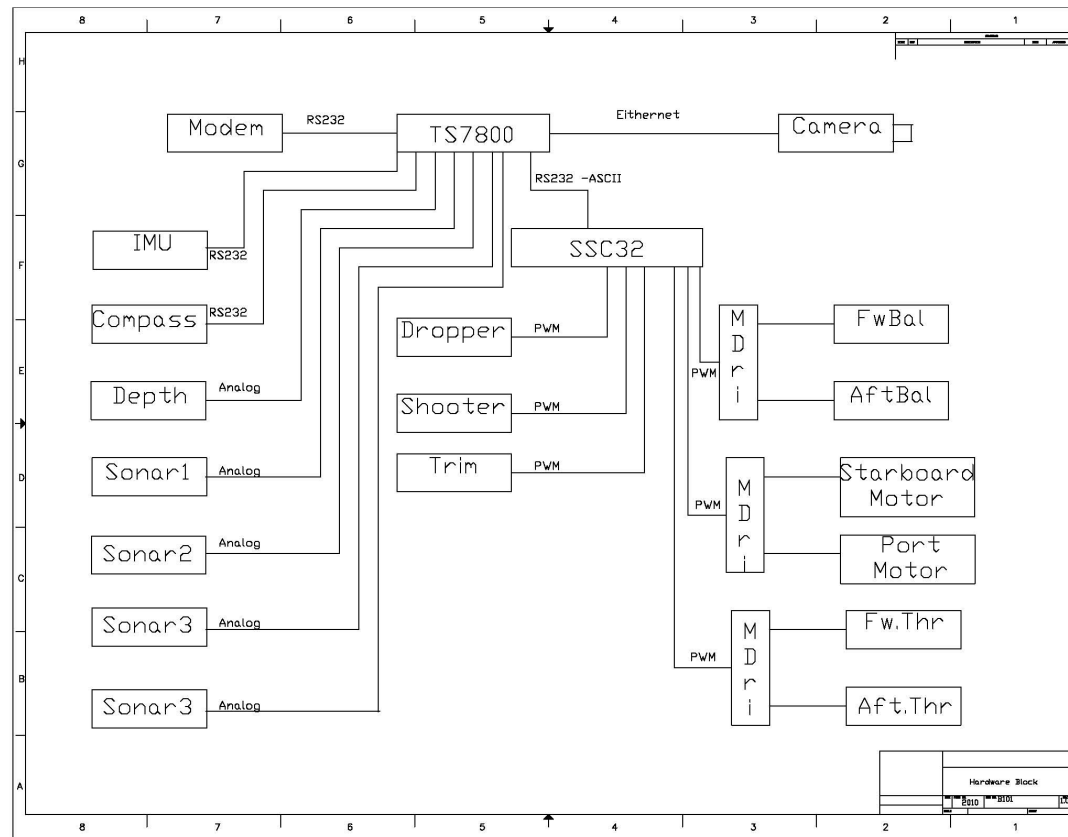
terminalSettings.c_cflag |= CREAD & ~CLOCAL;

```

```

// SN1
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag &= ~
terminalSettings.c_cflag |= CS8; // set for 8 bits.

```



AUV Block Diagram



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexc
```

```
#include "Serial
```

```
SerialInterface:
```

```
{
```

```
serialPort = c
```

```
if(serialPort
```

```
{
```

```
close(serial
```

```
throw std::r
```

```
);
```

```
struct termios
```

```
// get the cur
```

```
tcgetattr(seri
```

```
// set baud rate
```

```
cfsetispeed(&terminalSettings, (speed_t)baudRate);
```

```
cfsetospeed(&terminalSettings, (speed_t)baudRate);
```

```
terminalSettings.c_cflag |= CREAD & ~CLOCAL;
```

```
// SN1
```

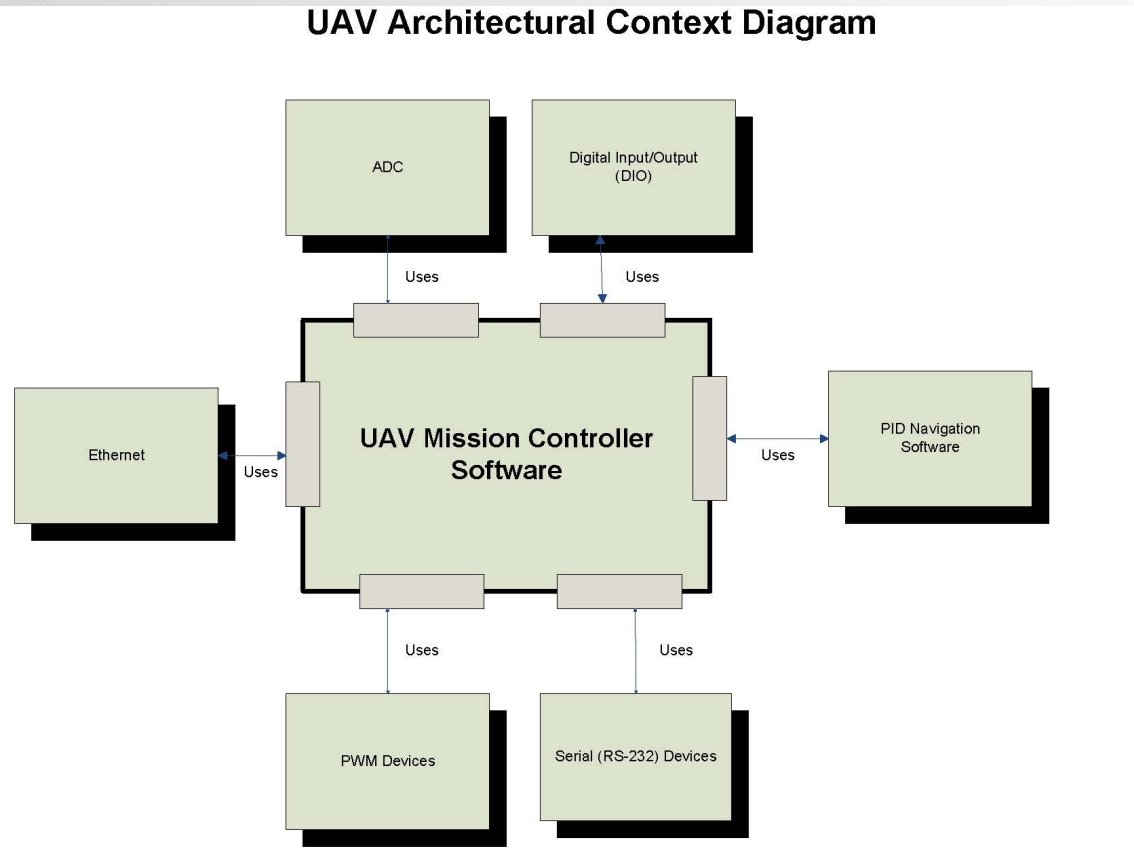
```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= CS8; // set for 8 bits.
```

UAV Architectural Context Diagram



AUV UML-Based Context Diagram




```

#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"

SerialInterface::SerialInterface()
{
    serialPort = open(deviceName, O_RDWR);

    if(serialPort <= 0)
    {
        close(serialPort);
        throw std::runtime_error("Failed to open serial port");
    }

    struct termios terminalSettings;
    tcgetattr(serialPort, &terminalSettings);

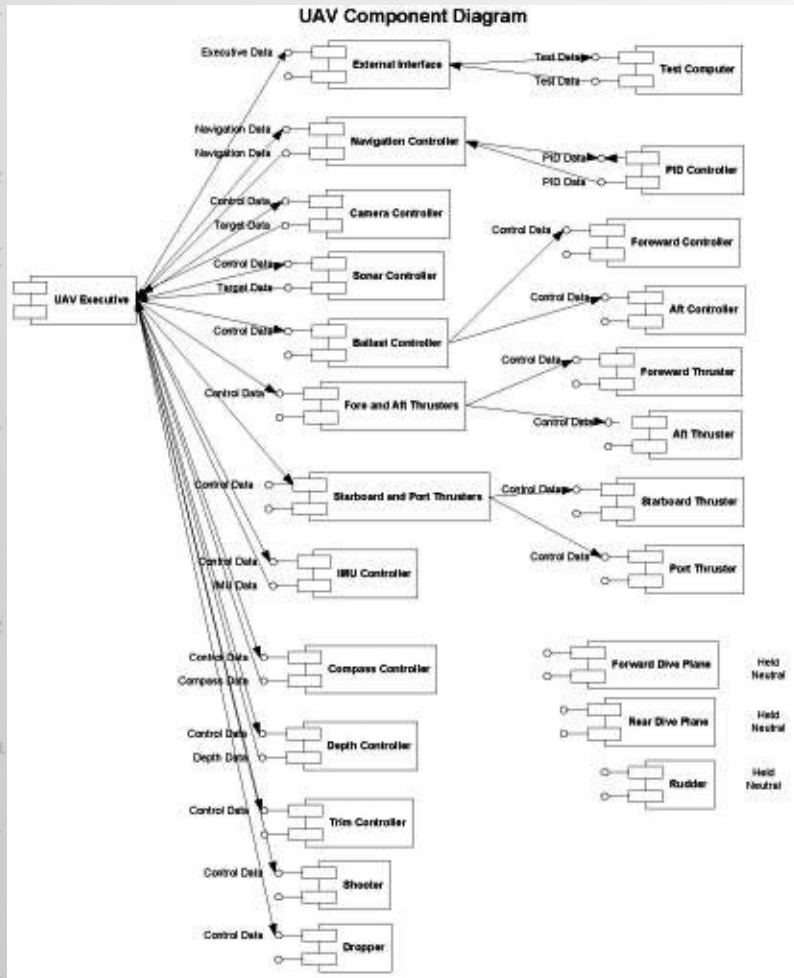
    // get the current set baud rate
    tcgetattr(serialPort, &terminalSettings);

    // set baud rate
    cfsetispeed(&terminalSettings, (speed_t)baudrate);
    cfsetospeed(&terminalSettings, (speed_t)baudrate);

    terminalSettings.c_cflag |= CREAD & ~CLOCAL;

    // SN1
    terminalSettings.c_cflag &= ~CS8;
    terminalSettings.c_cflag &= ~CSTOPB;
    terminalSettings.c_cflag &= ~CMASK;
    terminalSettings.c_cflag |= CS6; // set for 6 bits.
}

```



AUV UML-Based Component Diagram



```
#include <fcntl.h>
#include <sys/termios.h>
#include <iostream>
#include <stdexcept>

#include "SerialInterface.h"
```

```
SerialInterface::SerialInterface(std::string device, int baudRate)
```

```
{
```

```
ser
```

```
if (
```

```
{
```

```
c
```

```
t
```

```
str
```

```
//
```

```
tcg
```

```
// set baud rate
```

```
cfsetispeed(&terminalSettings, (speed_t)baudRate);
```

```
cfsetospeed(&terminalSettings, (speed_t)baudRate);
```

```
terminalSettings.c_cflag |= CREAD & ~CLOCAL;
```

```
// SN1
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag &= ~
```

```
terminalSettings.c_cflag |= CS8; // set for 8 bits.
```

University of West Florida School of Science and Engineering Unmanned Ground Systems/Tour Robot



University of
West Florida

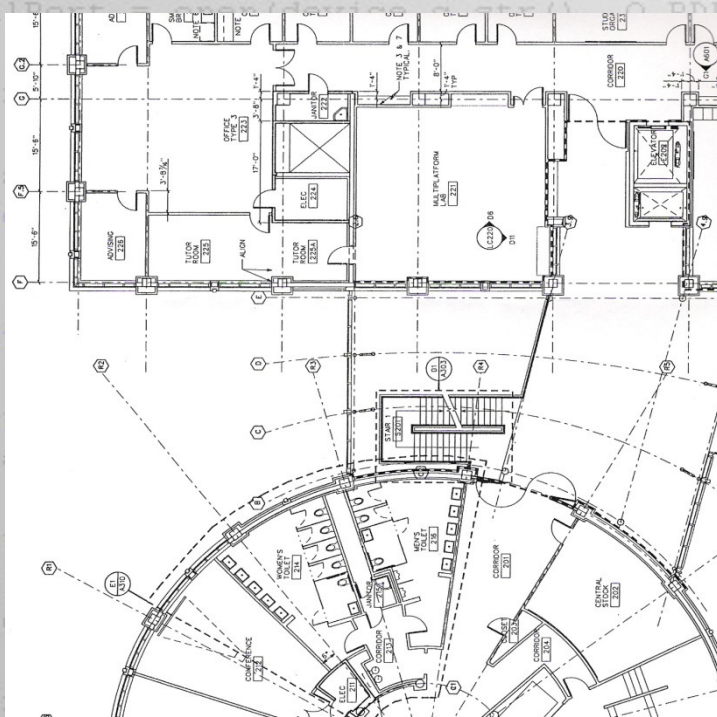
What is a Tour Robot!?

- Human-sized autonomous robot
- Intended to take prospective students on tours of the new state-of-the-art Science and Engineering building
- Currently being designed and built by UWF students
- Joint effort by the UWF Electrical & Computer Engineering, Computer Science and Art Departments



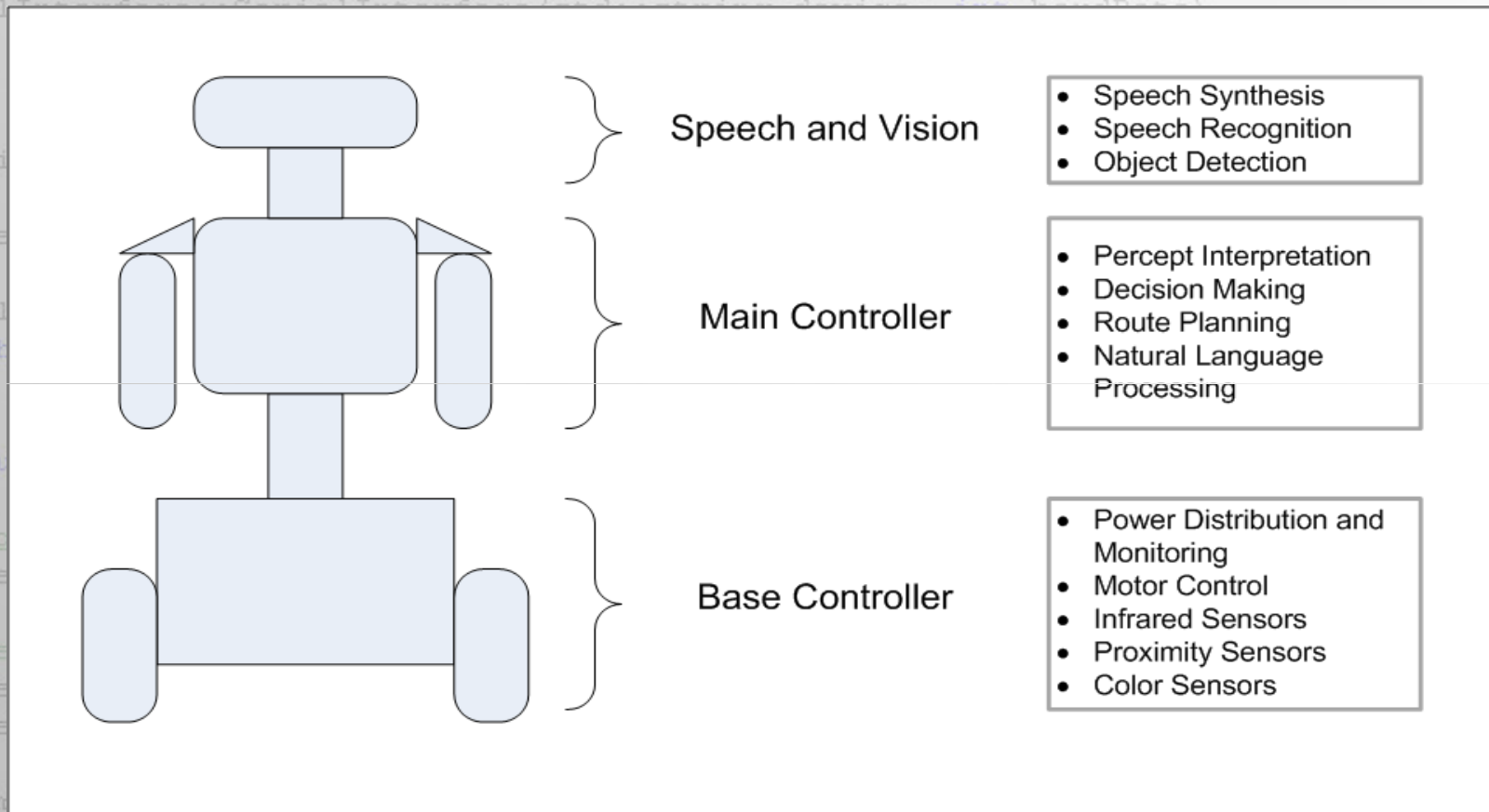
UWF Commitment to Tour Robot

Science and Engineering building considerations:



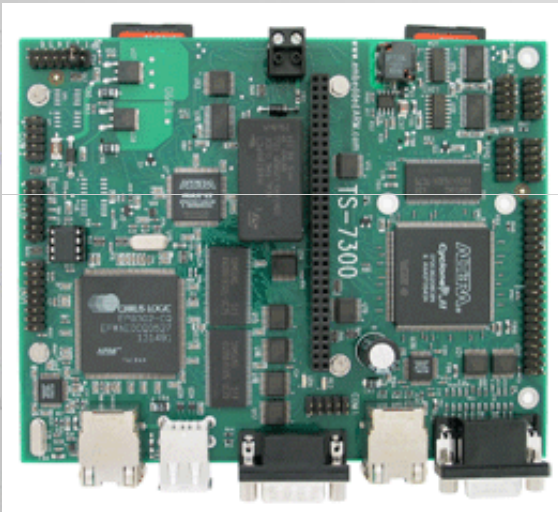
- Floor tile plan designed to accommodate robot navigation
- Solar panels on roof to provide all power for Tour Robot

High Level Design



Base Controller

Designed to provide the Main Controller with a generic interface to sensors and actuators.

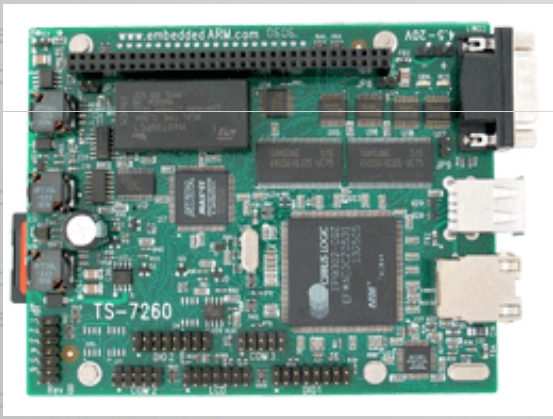


TS-7300 ARM-9 Embedded SBC

- Interfaces with various sensors (infrared, proximity, color, etc.) using PC/104 expansion bus, serial ports and digital I/O ports.
- Actuates motors using servo controller and motor drivers.
- Communicates with Main Controller using Ethernet communication.

Main Controller

Intended to perform more complex decision-making software operations requiring significant computation.



TS-7260 ARM-9 Embedded SBC

- Communicates with both the Base Controller and the Speech/Vision Controller to interpret the environment
- Searches for suitable sequences of actions that will achieve goals
- Responds to the environment by signaling the Base Controller and Speech/Vision Controller to activate actuators.

Speech/Vision Controller

Intended to provide an engaging machine-to-human interface.

Goals:

- Synthesize speech for leading tours
- Perform speech recognition to achieve an interactive experience
- Attempt to recognize movement of humans
- Attempt to detect and make note of objects within the environment



Learning Opportunities

- Practical, hands-on, team-oriented product development
- Embedded, real-time hardware and software development
- General robotics
- Network communication
- Sensor interfacing
- Intelligent Agents
- Speech and Image Processing
- Structural and aesthetic design
- Human-machine interface development



Questions?

```
#include <fcntl.h>
#ir
#ir
#ir
#ir

SerialInterface::SerialInterface(std::string device, int baudRate)
{
    serialPort = open(device.c_str(), O_RDWR | O_NOCTTY | O_NDELAY);

    if(serialPort <= 0)
    {
        // ...

        tcgetattr(serialPort, &terminalSettings);

        // set baud rate
        cfsetispeed(&terminalSettings, (speed_t)baudRate);
        cfsetospeed(&terminalSettings, (speed_t)baudRate);

        terminalSettings.c_cflag |= CREAD & ~CLOCAL;

        // SN1
        terminalSettings.c_cflag &= ~
        terminalSettings.c_cflag &= ~
        terminalSettings.c_cflag &= ~
        terminalSettings.c_cflag |= CS8; // set for 8 bits.
```

David Algeo

(850) 857-6411

dalgeo@uwf.edu

Shawn Adams

(850) 696-2390

adams@students.uwf.edu

Gretchen VanValkenburg

(850) 474-2875

gvalkenb@students.uwf.edu



**University of
West Florida**