We began work on the project in the fall semester, and much of it is complete at this point. My work has focused mainly on the design and implementation of electronic and software based control systems. The S-90 has three main methods of controls: joystick, RC, and steering wheel with pedals. These methods each control three main systems for the S-90: steering, throttle and brakes. Additionally a number of other systems go into making the S-90 safe and reliable.

The control methods for this project have received a lot of attention in the past months. As of right now, the joystick controls all three main systems enough for test driving to commence. The joystick being used for the S-90 outputs similar to a potentiometer. Two axis of movement are being used for control and each has their own output that is modified by moving the joystick. The steering wheel and pedals work much the same way, except that the throttle and brake pedals each have their own potentiometer. The analog voltage adjusted by the joystick is obtained by the PIC microcontroller via 8-bit analog to digital conversion. This conversion is based off of a 5V reference source and gives a number to the microcontroller between 0 and 255 based on level of the signal voltage compared to reference. This A/D count not only represents the position of a particular joystick axis, but also is very easy to work with in the context of the program. The software makes use of simple linear equations to adjust the input and create output information.

The three main systems being controlled by the software each function by receiving a PWM signal. The steering and braking motors both are run off of speed controllers which translate a PWM signal to the direction and level of voltage they deliver to the motor. The PWM signal is generated by the microcontroller and is a 50Hz signal. A pulse width of 1.5ms corresponds to a neutral motor. Full forward and full reverse correspond to pulse widths of 1ms and 2ms. The microcontroller PWM output to these speed controllers is all or nothing, since the speed controller does not determine rotational position. To find position, both the steering and braking systems have a separate transducer. In both cases the feedback comes in the form of an analog
voltage. The same method described above is used to obtain information about these voltages in the microcontroller. The software takes the feedback signal and compares it to the user control signal to determine which way to move the motor. The software routine updates every 20ms ensuring that changes to the input are serviced almost instantaneously.

Figure 1: Steering and braking software block diagram

The throttle system is simpler than the steering and braking, mainly due to the fact that it is simply a small servo that provides its own feedback. The servo has 90 degrees of rotation which correspond to a pulse width changing from 2ms to 1ms. The software is designed to take the A/D count reading from the user control and normalize it to its corresponding pulse width using a simple linear equation. Since the servo tracks its own position the microcontroller PWM output is a linear range of pulse widths instead of the all or nothing of the steering and braking.

Other components that are included in the software at this point are the speed indicator, speed governor, and the emergency shutdown routine. The speed indicator works by taking a signal from a switch that is mounted near the rear axle. The switch is closed once per revolution and this triggers an interrupt in the software. The software services the interrupt by incrementing a counter. Every 2 seconds this counter is checked and reset. The count obtained is then run through an equation to equate from rotations per two seconds to mph. The speed governor works with information from the speed indicator. When the speed reaches a set point the governor routine is entered and takes over throttle control from the user. It uses timing to rotate the throttle servo back to zero at a controlled rate. During this time the user does not control the throttle,
unless the user starts braking, in which case the throttle snaps back to zero. The speed governor routine continues until the speed indicator reads under the set point.

The emergency shutdown routine is an important safety feature for the S-90. It is essentially a one-button kill switch. When the routine is activated it immediately moves the throttle servo to the zero position. At the same time the routine fully applies the brakes. The engine is also grounded by the routine to ensure that it will not be applying power to the wheels in the event of an electrical failure.

![Development board with microcontrollers](image)

**Figure 2: Development board with microcontrollers**

The features described in this section are all software based. Most are fully completed as stated. The others are completed and just require calibration of the linear equations to match the actual physical characteristics of the controlled device.
**Future Work:**

The software for the S-90 is almost complete. The biggest obstacle is getting the remote control working well. At this point it has worked before, but is sporadic. The remote control is different from the other methods of control in that it is received at the microcontroller as a PWM signal instead of an analog voltage. This requires reconstruction of the PWM signal in the microcontroller using on-board timers. The method works, but calibration has been difficult without an ICD. This will be the focus for the immediate future. Additionally, safeties will be added to the other software routines allowing certain routines to only be accessed when the S-90 is stopped.

**Time Line:**

Jan 29th – Feb 5th: Get remote control working properly

Feb 5th – Feb 12th: Finish software and electronics

**Project Review:**

The S-90 is coming along fast with most of the systems done. The braking software needs to be calibrated to the mechanical system, but after that test driving with the joystick is quite feasible. Every system tested so far has worked and some have exceeded expectations. The project is on track to finish weeks ahead of the deadline and the final deliverable should meet every expectation and more.

**Hours Worked:** 130