# Table of Contents

Design Overview .................................................................................................................. 2  
Drive/Control Functions ..................................................................................................... 4  
Ball Collection Functions .................................................................................................. 6  
Ball Collection Continued ................................................................................................. 8  
Color Sort Functions .......................................................................................................... 11  
Ball Launch Functions ....................................................................................................... 17  
Conclusion .......................................................................................................................... 19  
Appendices .......................................................................................................................... 20  
  Appendix A: Photos ........................................................................................................... 20  
  Appendix B: Components ................................................................................................. 29  
  BX-24P Microcontroller ..................................................................................................... 29  
  Sharp Infrared Sensor ....................................................................................................... 29  
  Jaguar Motor Controller ................................................................................................... 29  
  Servo ................................................................................................................................. 29  
  Solenoid ............................................................................................................................. 29  
  ALPEC-Team Laser Diode ............................................................................................... 29  
  LED Strip ........................................................................................................................... 29  
  Battle Switch .................................................................................................................... 29  
  Futaba Remote Controller ............................................................................................... 29  
  Futaba Receiver ............................................................................................................... 30  
  Limit Switch ..................................................................................................................... 30  
  Top 24” Throw Actuator ................................................................................................. 30  
  Front 4” Throw Actuator ................................................................................................. 30  
  Fan ..................................................................................................................................... 30  
  Appendix C: SolidWorks CAD Renders ............................................................................ 31  
  Appendix D: BasicX (BX24P) Code .................................................................................. 33  
    Primary.bas ..................................................................................................................... 33  
    Secondary.bas ................................................................................................................. 36  
  Appendix E: Wiring Diagrams .......................................................................................... 42  
    SAMSON Complete Wiring Diagram ........................................................................... 42  
    High Current Wiring ....................................................................................................... 43  
  Appendix F: Project Timeline ......................................................................................... 44  
  Appendix G: Bill of Materials ......................................................................................... 45  
  Works Cited ....................................................................................................................... 46
Design Overview

The Semi-Autonomous Marauder with Sensor Optimized Navigation (SAMSON) was constructed with four goals in mind: simplicity, robustness, reliability, and task efficiency. Careful design processes and independent system testing ensured that each team’s components were as simple, reliable, sturdy, and efficient as possible. After extensive bench testing, components were integrated and refined to create SAMSON.

SAMSON addresses the “Battle Hoops” challenge by first collecting ping pong balls located both on and off the contest table, autonomously sorting them according to color, and launching them into their designated hoop by the operator. The remainder of this paper and the accompanying flowchart (Figure 1) present an overview of the project as a whole and address each system and function in further detail.
Figure 1: SAMSON Overview Flowchart
**Drive/Control Functions**

SAMSON is remotely powered by a Futaba remote control device. (Components – Futaba Remote Controller) It also has an autonomous control mode that handles the navigation to the table in the center of the arena. In either mode, SAMSON is driven by two high-torque wheelchair motors positioned perpendicular to the chassis. The motors are positioned this way to allow an open internal chassis for mounting all its required components. Our design called for each subsystem on the robot to be mounted inside the chassis in order to promote a simpler and smaller design envelope. Access panels were also incorporated to allow for easy repair and maintenance of all subsystems (See Figure 2). The motors were mounted outside to present the drivetrain as its own external feature. With the motors being located outside the chassis, proper chain tension and chain alignment could be achieved with the help of custom adjustable motor mounts.

The entire drivetrain configuration is similar to that of a tank; both wheels on each side move at the same time and in the same direction as the motor. A tank-like configuration provides four wheel drive traction and stability while in motion. This gives the advantage of all four wheels rolling at the same time and at the same rate. Other
advantages of a tank-like drivetrain are a zero turning radius, and the ability to rotate 360 degrees from the center of the chassis. Our identical wheelchair motors are controlled by individual Jaguar motor controllers (See Appendix B). Each motor controller has the ability to “ramp up” the power to our motors, which allows the operator to make sure the control isn’t “choppy.”

A clean start up and shut down of each motor makes it easier for the robot to navigate towards the table in autonomous mode. Each drive motor is controlled via tele-operation from a Futaba remote control, which sends signals to a Futaba receiver. (See Appendix B) Those signals pass through a switch and then into a Jaguar motor controller. (See Appendix B) The switch after the receiver acts as the director, it directs the signal straight to the motor controller for tele-operation, or it directs the signal to the microcontroller for autonomous mode. (See Appendix B) Autonomous mode is required for the robot to enter the circle unassisted. To help the operator line up from outside the circle before entering, laser diodes are used to provide the operator with visual feedback. (See Appendix B) The visual feedback of each laser diode informs the operator of the angle at which the robot is approaching an object. Once the robot has reached an appropriate position, the operator signals the microcontrollers to begin autonomous mode by toggling the autonomous switch.

Once SAMSON enters the circle, and begins its autonomous subroutines, it relies solely on sensor data to make decisions on how to locate and approach the table. Sharp IR sensors and a single limit switch were used to locate the table. (See Appendix B) The essential function of the IR sensors is to determine the best course of action to
reach the table, (adjusting either right or left) while the limit switch indicates that SAMSON has successfully made contact with the table. A signal is then sent to the microcontroller (BX-24P) to begin the ball collection process.

**Ball Collection Functions**

The next step in the autonomous process is the first of two ball collecting routines. Once the limit switch detects the table, the top linear actuator extends the collection hood out. (See Appendix B) After the collection hood is fully extended, the front linear actuators are powered to lift the table off the ground 4 inches to optimal collection height. (See Appendix B) After the table is raised, a limit switch will trigger and signal the top actuator to begin retract the collector hood back. While the collector hood retracting, the balls on top of the table are pushed into our vacuum-formed ball sorting storage container. This element is a custom design which was vacuum-formed specifically for SAMSON. When the collector hood is fully retracted, the front actuators lower the table and then SAMSON backs out of the circle. Autonomous mode is then switched off by the operator and tele-operation can resume. Figure 3 shows the finished autonomous ball collection system.
Figure 4
Autonomous Control Flowchart
Ball Collection Continued

Once the switch is toggled and autonomous mode is deactivated, the second of the two ball collection functions can begin. The tele-operated ball function is designed for balls located on the ground. The operator will locate the balls scattered on the ground, and will simply drive over them picking up the balls. A paddle sweeps across the ping pong balls and coerces them into a collection tray. Figure 3 shows the paddle, which was manufactured by machining a spindle (to act as an axle), through the inside of a PVC pipe. Foam paddles were then bolted to the spindle to make contact with the idle balls and send them into the tray. The transmission that drives the paddle was calculated using the formula presented by input movement/output movement. The factor used by our design is 0.5, which allows an optimization of torque to speed to power usage. The main shape of this tray was printed in the 3D printer, because of the precision required in the inclined planes (Greenemeier, 2013). The design incorporates two gravity actuated planes which work in opposite directions. The interface of these two planes hinders the balls from crossing from one plane to the next, and the balls are forced over this junction by the foam paddle. The final effect is that the balls are driven up the first plane and across the uneven interface. At this point, gravity draws the balls down the second plane and into the elevator, which lifts the balls up and dumps them into the ball sorting storage container. The Tele-operated flowchart (See Figure 6) shows this process in detail.
If the balls are not all collected, the operator locates the rest of the balls and recover accordingly. Once it is determined all of the balls are in the ball sort storage container, the next process can begin.

Figure 5
Loose Ball Collection System
Figure 6
Tele-Operated Ball Recovery Flowchart
Color Sort Functions

As soon as the balls are retrieved from the table by the drawer and fork collection system, they must be sorted by color. Because loose balls are strewn across the floor will be coming in constantly, the sorting will begin before the balls are completely collected. The Color Organizing and Sorting Optical Device (COSOD) is mounted directly under the ball storage container, that the device is constantly fed balls due to the force of gravity as seen in figure. The COSOD is a mechanism designed, in SolidWorks and fabricated in house, to address the issue of sorting the balls by color. The COSOD’s main parts (the rotor and the base) were printed on the Objet Alaris 3D printer (Greenemeier, 2013). See Appendix C for display of the 3D printed parts.

The rotor, powered by a modified servo motor, spins inside the base on a steel axle. Servos are positioned in a configuration that allows for a maximum rotation of approximately 120 degrees. Standard servos must be modified to run continuously by adding multiple hardwired resistors in place of the rotational potentiometer, which must be physically altered to allow for unhindered rotation. This spinning of the rotor drags the balls through the
sensing ranges of the four optical color sensors. The optical color detection is handled by four Allen Bradley 45 CLR color sensors (See Figure 8), while an Allen Bradley Hall effect sensor is used to ensure the balls are aligned in the gate. There are small pieces of metal embedded into the side of the rotor, which are precisely placed so the Hall effect sensor detects them as the rotor is properly aligned within the base. Once the sensor detects the color it seeks and the Hall Effect sensor is lined up, the gate below the ball opens. If the ball color is not what the color sensor seeks, the gate corresponding to that optical color sensor will remain closed. The gating system consists of 8 gates, which are uniquely designed to allow only one ball to drop into a tube at a time, and 4 servo motors. The design incorporates a split-gate mechanism which allows the top and bottom gates to be separated by 42.5mm and rotated 90 degrees apart. Figure 9 illustrates the COSOD functional flowchart. The balls are dropped from the gate into one of four color coded pipes.
Figure 9
Color-sorting Process Flowchart
Although they appear to simply transport the balls to the loading device, the four pipes between the color sorted and the firing mechanism were specifically designed for two functions: transport and storage. Because the four pipes are angled downhill along their entire path, gravity can be used to transport the balls to the loading mechanism. Although some portions of the system are steeper, the minimum slope is approximately 5 degrees. This allows for a slight yet constant force to load the balls into the barrel of the firing mechanism. Due to the change in rules for the competition, and the consequent reduction of the total number of balls, the design of the transport and storage system was changed. The initial dimensions for the system required the pipes be 25 inches long to accommodate the maximum number of balls in a worst-case scenario. Under the new rules, that dimension could be reduced to 17 inches, saving valuable robotic real estate.

Once the balls reach the loading mechanism, they are stopped by a solid gate which doubles as a seal. Obviously the gate stops the balls from running into the barrel and causing random misfire. However, it also seals the barrel while other colors are firing. This seal is not required to be air tight, because the purpose is only to force most of the air through the barrel, instead of sideways into the transportation and magazine (storage) area.

The stop gates are controlled by a microprocessor that automates the complete firing process. The operator selects a color and then activates a “Fire” button. This sends two of five signals to the firing processor. The first is a color indication signal, which selects the color to be fired, and the second is the actual trigger pulse. Although
this could’ve been accomplished with one signal, the separation allows the operator to load one ball before the robot is even in position to fire that ball. For example, a signal for a red ball could be immediately followed by a fire pulse. In this event, the processor would begin the steps required to fire one red ball. The first step is to ensure the fan gate is closed. Where the air to be flowing as the balls is loaded, the air would escape through the seal gate and stopping the loading process. This would be similar to loading a musket while it was firing a gunpowder charge. With our system, the failure would be much less violent, but the analogy is effective. The fan gate must be closed before any ball can be loaded. As soon as the fan is ready, one of the seal gates can be opened. This will allow a ball to roll into the barrel. A single solenoid extends to prevent the second ball from also rolling into the barrel. The seal gate and the solenoid work inversely so that only one ball is sent at a time. The normal position for the solenoid is open while the normal position for the seal gate is closed. This allows one ball to roll past the solenoid but not past the seal gate. When that ball is to be loaded, the solenoid closes as the gate opens, hindering the secondary balls from moving while the gate is open. Once the primary ball is loaded and the gate seals again, the solenoid can return to its normally open position. This allows the secondary ball to roll forward 42.5 millimeters where it becomes the primary ball for the subsequent firing round. (42.5 millimeters is the diameter of a standard Ping-Pong ball plus ~5% clearance)

After the ball is loaded into the barrel chamber, the fan gate can open and allow the airflow from the fan to propel the ball out of the barrel and into the net. This sequence is timed for its operation as no sensors detect the position of the ball inside the barrel. This is necessary because the volume of the inside of the barrel is used as a
pressure chamber to propel the ball forward at the optimal speed (Fitzgerald, 2003). Adding sensors would have reduced the efficiency and reliability of this firing system. Since no sensors were used, the firing processor must approximate the time required for the ball to travel between the chambers of the barrel to the point where the airflow ceases to affect the flight. This is accomplished with a simple time delay of one second. Although the process is slightly shorter, the selected delay performs within the system reliably.

The design of the barrel required much consideration of frictional forces within the tube. The final velocity is limited by the tubing length because of the acceleration of the ball along the tube (Mahazzabi, 2002).

As soon as the ball exits the muzzle, the microprocessor closes the fan gate and prepares to load the subsequent ball. If the processor is already receiving a color indication signal, it would load a ball of the indicated color, and then wait for the fire pulse. This completes the loop of the firing automation processor. The loop will be repeated until all of the collected balls are in the baskets.
Ball Launch Functions

Once the balls are sorted into each of their storage tubes, ball launching can begin. When the COSOD recognizes it’s finished, two three-ampere fans turn on to prepare for the firing of the ping pong balls. Before shooting takes place, the operator of SAMSON will drive to the closest colored hoop. Two Alpec-Team Inc. lasers are used to ensure that the operator positions SAMSON into the most optimal position to launch the balls. Once in position, the operator will toggle the corresponding switch to fire one ball of appropriate color into the hoop. The firing cycle begins when a solenoid is triggered to let one ball drop down into the loading chamber while simultaneously rotating a butterfly valve to plug the fans, preventing airflow in the firing chamber. (See Appendix B) From there, the ball is met with another gate that is on a time delay in the loading chamber. When the time delay expires, the gate will open and a split-second later the gate to cap the fan is released and the firing chamber is then filled with rushing air, forcing the ball into the barrel and launching it. The entire firing cycle is looped until the operator toggles the switch again. This makes the firing cycle automatic. In the event that the first ball does not make it into the hoop, the operator has two options: reposition SAMSON while the firing cycle continues, or stop the firing cycle by toggling the switch on the remote, reposition SAMSON, and then toggle the switch again to restart the firing cycle (Gardner, 1999). Figure 10 shows the fan and launching barrel.
Figure 11
Ball-launching Flowchart
Conclusion

SAMSON highlights integration of several independent and complex systems into one unified platform. Robustness in design and implementation has been achieved through careful materials selection, as well as careful assembly. This ensures the chassis can more than accommodate for external forces while in operation. Reliability has been achieved through thorough testing (ensuring SAMSON’s results are repeatable) and thorough documentation on each system to aid in debugging and maintenance. Simplicity has been achieved through efficient design. Every system was incrementally engineered to decrease the amount of individual components and moving parts, as fewer parts leads to unforeseen issues. SAMSON’s design was entirely task oriented at each step in the process, and every part incorporated onto the chassis has a purpose and usefulness to the system as a whole.
Appendices

Appendix A: Photos

Foundation Work on the Chassis

Figure 8

Prototype Hopper
Testing Motor Controllers

Loose Ball Recovery Elevator
Rapid Prototype of Loose Ball Recovery Tray

Bench Test of Ball Launch System
Post Color-Sort Storage Pipes

Color Sorter Wiring and Motor Assembly
Fork Lift Actuators

Testing LED Strips Within Wiring
BX-24P Micro-Controllers

Completed Bench Test Wiring
Assembled Color Sorter to Storage Pipes

Side Panel Circuit Mounts
Loose Ball Recovery Paddle Drive System

Wiring SAMSON
Forklift Limit Switch Integration
Appendix B: Components

BX-24P Microcontroller
Documentation available at:

http://www.basicx.com/Products/BX-24/bx24overview.htm
http://www.basicx.com/Products/BX-24/bx24specs.htm

Sharp Infrared Sensor
Documentation available at:

http://www.sharpsma.com/webfm_send/1489

Jaguar Motor Controller
Documentation available at:


Servo
Documentation unavailable.
Manufacturer's page: http://www.tacticrc.com/servos/index.html

Solenoid
Documentation available at:

http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Robotics/ZHO-420S.pdf

ALPEC-Team Laser Diode
Documentation available at:


LED Strip
Documentation available at:


Battle Switch
Documentation available at:

http://www.dimensionengineering.com/products/battleswitch

Futaba Remote Controller
Documentation available at:

Futaba Receiver
   Documentation available at:
   
   http://www.gpdealera.com/cgi-bin/wgainf100p.pgm?I=FUTL7645

Limit Switch
   Documentation unavailable.
   
   Limit switches were gleaned from a previous project.

Top 24” Throw Actuator
   Documentation available at:
   

Front 4” Throw Actuator
   Documentation available at:
   

Fan
   Documentation unavailable.
   
   Fans were gleaned from a production server blade.
Appendix C: SolidWorks CAD Renders

Color Sort Base and Gates

Color Sort Base, Rotor, and Gates
Ground Ball Collection Tray

Ground Ball Collection Tray Half
Appendix D: BasicX (BX-24P) Code

Primary.bas

Option Explicit

' Pin constants
Public Const LeftForward As Byte = 13
Public Const LeftReverse As Byte = 14
Public Const RightForward As Byte = 15
Public Const RightReverse As Byte = 16
Public Const TableDetection As Byte = 18
Public Const LeftMotor As Byte = 8
Public Const RightMotor As Byte = 7
Public Const Actuator1 As Byte = 9
Public Const Actuator2 As Byte = 10

' Main subroutine
Public Sub Main()
    Dim AutoSwitch As Byte
    Dim MultiSwitch As Integer
    Call PutPin(25,0)
    Call PutPin(26,1)
    Call PutPin(1,0)
    Do
        ' Check for autonomous switch
        AutoSwitch = GetPin(5)
        If (AutoSwitch = 1) Then
            Call PutPin(25,1)
            Call PutPin(26,0)
            Call Auto()
        End If
        Call Delay(0.02)
    Loop
End Sub

' Autonomous subroutine
Public Sub Auto()
    Dim AutoSwitch As Byte
    Dim LeftForwardReading As Byte
    Dim LeftReverseReading As Byte
    Dim RightForwardReading As Byte
    Dim RightReverseReading As Byte
    Dim TableDetectionReading As Byte
    Do
        ' Check for autonomous switch
        AutoSwitch = GetPin(5)
        If (AutoSwitch = 0) Then
Call PutPin(25,0)
Call PutPin(26,1)

Exit Sub

End If

LeftForwardReading = GetPin(13)
LeftReverseReading = GetPin(14)
RightForwardReading = GetPin(15)
RightReverseReading = GetPin(16)
TableDetectionReading = GetPin(18)

' If leftforward and not leftreverse
If (LeftForwardReading = 1) And (LeftReverseReading = 0) Then
    Call MoveLeftForward()
ElseIf (LeftForwardReading = 0) And (LeftReverseReading = 1) Then
    Call MoveLeftReverse()
End If

' If rightforward and not rightreverse
If (RightForwardReading = 1) And (RightReverseReading = 0) Then
    Call MoveRightForward()
ElseIf (RightForwardReading = 0) And (RightReverseReading = 1) Then
    Call MoveRightReverse()
End If

' If detecting table
If (TableDetectionReading = 1) Then
    Call PutPin(1,1)
    Do
        Call Delay(0.02)
    Loop
End If

Call Delay(0.02)
Loop

End Sub

' Move forward
Public Sub MoveForward()
    Call MoveRightForward()
    Call MoveLeftForward()
End Sub

' Move back
Public Sub MoveReverse()
    Call MoveRightReverse()
Call MoveLeftReverse()

End Sub

' Rotate left (zero turn)
Public Sub MoveRotateLeft()
    Call MoveRightForward()
    Call MoveLeftReverse()
End Sub

' Rotate right (zero turn)
Public Sub MoveRotateRight()
    Call MoveLeftForward()
    Call MoveRightReverse()
End Sub

' Move right and go forward
Public Sub MoveRightForward()
    Call PulseOut(RightMotor, 0.00191, 1)
End Sub

' Move left and go forward
Public Sub MoveLeftForward()
    Call PulseOut(LeftMotor, 0.00191, 1)
End Sub

' Move right and go backward
Public Sub MoveRightReverse()
    Call PulseOut(RightMotor, 0.00091, 1)
End Sub

' Move left and go backward
Public Sub MoveLeftReverse()
    Call PulseOut(LeftMotor, 0.00091, 1)
End Sub
Option Explicit

' Pin constants
Public Const LeftForward As Byte = 6
Public Const LeftReverse As Byte = 7
Public Const RightForward As Byte = 8
Public Const RightReverse As Byte = 9
Public Const TableDetection As Byte = 10
Public Const MiddleRangeLeftSensor As Byte = 13
Public Const MiddleRangeCenterSensor As Byte = 14
Public Const MiddleRangeRightSensor As Byte = 15
Public Const LongRangeSensor As Byte = 16
Public Const CloseRangeLeftSensor As Byte = 18
Public Const CloseRangeRightSensor As Byte = 19
Public Const GreenLED As Byte = 25
Public Const RedLED As Byte = 26

' Main subroutine
Public Sub Main()
    Dim AutoSwitch As Byte
    Call PutPin(25,0)
    Call PutPin(26,1)

    ' Start table detection
    Call PutPin(TableDetection, 0)

    Do
        AutoSwitch = GetPin(5)
        If (AutoSwitch = 1) Then
            Call PutPin(25,1)
            Call PutPin(26,0)
            Call ApproachTable()
        End If
        Call Delay(0.02)
    Loop
End Sub

' Autonomous behaviour to approach the table
Public Sub ApproachTable()
    Dim CloseRangeLeftReading As Integer
    Dim CloseRangeRightReading As Integer
    Dim MiddleRangeLeftReading As Integer
    Dim MiddleRangeCenterReading As Integer
    Dim MiddleRangeRightReading As Integer
    Dim LongRangeReading As Integer
    Dim AutoSwitch As Byte
Do
' Check for autonomous switch
AutoSwitch = GetPin(5)
If (AutoSwitch = 0) Then
    Call PutPin(25,0)
    Call PutPin(26,1)
    Exit Sub
End If
' Get readings from IR sensors
CloseRangeLeftReading = GetIRReading(CloseRangeLeftSensor)
CloseRangeRightReading = GetIRReading(CloseRangeRightSensor)
MiddleRangeLeftReading = GetIRReading(MiddleRangeLeftSensor)
MiddleRangeCenterReading = GetIRReading(MiddleRangeCenterSensor)
MiddleRangeRightReading = GetIRReading(MiddleRangeRightSensor)
LongRangeReading = GetIRReading(LongRangeSensor)
' Range testing
If (MiddleRangeLeftReading > 200) And (MiddleRangeRightReading < 200) Then
    Call MoveRightForward()
    Call MoveStopLeft()
ElseIf (MiddleRangeLeftReading < 200) And (MiddleRangeRightReading > 200) Then
    Call MoveLeftForward()
    Call MoveStopRight()
ElseIf (MiddleRangeLeftReading < 200) And (MiddleRangeRightReading < 200) Then
    If (LongRangeReading > 400) Then
        Call MoveForward()
    Else
        Call MoveStopAll()
        Call SearchForTable()
    End If
Else
    Call MoveForward()
End If
End If
If (CloseRangeLeftReading > 400) And (CloseRangeRightReading > 400) Then
    Call FoundTable()
End If
Call Delay(0.02)
Loop
End Sub
' Adjust steering to find the table
Public Sub SearchForTable()
    Dim LongRangeReading As Integer
Dim MiddleRangeLeftReading As Integer
Dim MiddleRangeCenterReading As Integer
Dim MiddleRangeRightReading As Integer

Dim AutoSwitch As Byte
Dim SweepCounter As Integer
SweepCounter = 24

Do
    ' Check for autonomous switch
    AutoSwitch = GetPin(5)
    If (AutoSwitch = 0) Then
        Call PutPin(25, 0)
        Call PutPin(26, 1)
        Exit Sub
    End If

    ' Perform sweeps
    Do
        ' Check for autonomous switch
        AutoSwitch = GetPin(5)
        If (AutoSwitch = 0) Then
            Call PutPin(25, 0)
            Call PutPin(26, 1)
            Exit Sub
        End If

        ' Get readings from the IR sensors
        MiddleRangeLeftReading = GetIRReading(MiddleRangeLeftSensor)
        MiddleRangeCenterReading = GetIRReading(MiddleRangeCenterSensor)
        MiddleRangeRightReading = GetIRReading(MiddleRangeRightSensor)
        LongRangeReading = GetIRReading(LongRangeSensor)

        ' Judge distance
        If (LongRangeReading > 430) Or (MiddleRangeLeftReading > 200) Or (MiddleRangeCenterReading > 200) Or (MiddleRangeRightReading > 200) Then
            Call MoveStopAll()
            Exit Sub
        End If

        ' Rotate right
        Call MoveRotateRight()
        SweepCounter = SweepCounter + 1
        Call Delay(0.02)
    Loop Until (SweepCounter >= 50)
SweepCounter = 0

Do
    ' Check for autonomous switch
AutoSwitch = GetPin(5)

If (AutoSwitch = 0) Then
    Call PutPin(25,0)
    Call PutPin(26,1)
    Exit Sub
End If

' Get readings from the IR sensors
MiddleRangeLeftReading = GetIRReading(MiddleRangeLeftSensor)
MiddleRangeCenterReading = GetIRReading(MiddleRangeCenterSensor)
MiddleRangeRightReading = GetIRReading(MiddleRangeRightSensor)
LongRangeReading = GetIRReading(LongRangeSensor)

' Judge distance
If (LongRangeReading > 430) Or (MiddleRangeLeftReading > 200) Or (MiddleRangeCenterReading > 200) Or (MiddleRangeRightReading > 200) Then
    Call MoveStopAll()
    Exit Sub
End If

' Rotate left
Call MoveRotateLeft()
SweepCounter = SweepCounter + 1
Call Delay(0.02)
Loop Until (SweepCounter >= 50)
SweepCounter = 0
Loop

End Sub

' Return when the table is found
Public Sub FoundTable()
    Call PutPin(TableDetection, 1)
    Do
        ' All done here
        ' Behavior loops infinitely until autonomous behaviour switched off
        Call Delay(0.02)
    Loop
    Call Delay(0.02)
End Sub

' Return the reading from an Infared sensor
' Takes 10 readings
Public Function GetIRReading(ByVal Sensor As Byte) As Integer
    Dim i As Integer
    Dim Average As Integer
    Average = 0
    For i = 0 To 9
Average = Average + GetADC(Sensor)

Next
GetIRReading = Average \ 10

End Function

' Inch forward
Public Sub MoveForward()
    Call MoveRightForward()
    Call MoveLeftForward()
End Sub

' Inch in reverse
Public Sub MoveReverse()
    Call MoveRightReverse()
    Call MoveLeftReverse()
End Sub

' Rotate the robot left slightly (zero turn)
Public Sub MoveRotateLeft()
    Call MoveRightForward()
    Call MoveLeftReverse()
End Sub

' Rotate the robot right slightly (zero turn)
Public Sub MoveRotateRight()
    Call MoveLeftForward()
    Call MoveRightReverse()
End Sub

' Move forward and right
Public Sub MoveRightForward()
    Call PutPin(RightForward, 1)
    Call PutPin(RightReverse, 0)
End Sub

' Move forward and left
Public Sub MoveLeftForward()
    Call PutPin(LeftForward, 1)
    Call PutPin(LeftReverse, 0)
End Sub

' Move back and right
Public Sub MoveRightReverse()
    Call PutPin(RightForward, 0)
    Call PutPin(RightReverse, 1)
End Sub

' Move back and left
Public Sub MoveLeftReverse()
    Call PutPin(LeftForward, 0)
Call PutPin(LeftReverse, 1)

End Sub

' Stop moving left
Public Sub MoveStopLeft()
    Call PutPin(LeftForward, 0)
    Call PutPin(LeftReverse, 0)
End Sub

' Stop moving right
Public Sub MoveStopRight()
    Call PutPin(RightForward, 0)
    Call PutPin(RightReverse, 0)
End Sub

' Stop in all directions
Public Sub MoveStopAll()
    Call PutPin(LeftForward, 0)
    Call PutPin(LeftReverse, 0)
    Call PutPin(RightForward, 0)
    Call PutPin(RightReverse, 0)
End Sub
Appendix E: Wiring Diagrams

SAMSON Complete Wiring Diagram
### Appendix F: Project Timeline

<table>
<thead>
<tr>
<th>Task Mode</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
<th>Resource Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Project Timeline</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chassis Development and Control</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Brainstorm Design</td>
<td>4 days</td>
<td>Thu 9/5/13</td>
<td>Tue 9/10/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Final Design Completed</td>
<td>3 days</td>
<td>Tue 9/10/13</td>
<td>Thu 9/12/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Materials list Ready to order</td>
<td>4 days</td>
<td>Thu 9/10/13</td>
<td>Fri 9/13/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Install Table Ball Removal Equipment</td>
<td>25 days</td>
<td>Mon 9/16/13</td>
<td>Fri 10/18/13</td>
<td>3,4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Color / Ball Sorter</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Basic Design Completed</td>
<td>4 days</td>
<td>Thu 9/5/13</td>
<td>Tue 9/10/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Final Design Completed</td>
<td>2 days</td>
<td>Wed 9/11/13</td>
<td>Thu 9/12/13</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Materials list Ready to order</td>
<td>6 days</td>
<td>Fri 9/13/13</td>
<td>Fri 9/20/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Construction of Ball Sorter</td>
<td>7 days</td>
<td>Mon 9/23/13</td>
<td>Tue 10/13/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Testing and Adjustments</td>
<td>9 days</td>
<td>Thu 10/13/13</td>
<td>Tue 10/15/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Ball Launcher</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Basic Design Completed</td>
<td>4 days</td>
<td>Thu 9/5/13</td>
<td>Tue 9/10/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Final Design Completed</td>
<td>2 days</td>
<td>Wed 9/11/13</td>
<td>Thu 9/12/13</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Materials list Ready to order</td>
<td>3 days</td>
<td>Tue 9/17/13</td>
<td>Thu 9/19/13</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Construction of Ball Launcher</td>
<td>7 days</td>
<td>Mon 9/23/13</td>
<td>Tue 10/13/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Testing and Adjustments</td>
<td>12 days</td>
<td>Wed 10/13/13</td>
<td>Thu 10/17/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Semi-Autonomous Control</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Basic Design Completed</td>
<td>4 days</td>
<td>Thu 9/5/13</td>
<td>Tue 9/10/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Final Design Completed</td>
<td>2 days</td>
<td>Wed 9/11/13</td>
<td>Thu 9/12/13</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Materials list Ready to order</td>
<td>6 days</td>
<td>Fri 9/13/13</td>
<td>Fri 9/20/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Construction/programming of Semi Auto Components</td>
<td>11 days</td>
<td>Mon 9/23/13</td>
<td>Mon 10/7/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Testing and Adjustments</td>
<td>12 days</td>
<td>Tue 10/13/13</td>
<td>Wed 10/23/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Loose Ball Recovery</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Basic Design Completed</td>
<td>4 days</td>
<td>Thu 9/5/13</td>
<td>Tue 9/10/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Final Design Completed</td>
<td>2 days</td>
<td>Wed 9/11/13</td>
<td>Thu 9/12/13</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Materials list Ready to order</td>
<td>6 days</td>
<td>Fri 9/13/13</td>
<td>Fri 9/20/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Construction of Loose Ball Recovery Equipment</td>
<td>11 days</td>
<td>Mon 9/23/13</td>
<td>Mon 10/7/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Testing and Adjustments</td>
<td>3 days</td>
<td>Fri 10/13/13</td>
<td>Tue 10/15/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>INTEGRATION</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Integrate/Power Drawer</td>
<td>5 days</td>
<td>Mon 10/21/13</td>
<td>Wed 10/23/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Integrate Hopper / Ball Sorter</td>
<td>2 days</td>
<td>Wed 10/16/13</td>
<td>Thu 10/17/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Install Ball Shooter</td>
<td>1 day</td>
<td>Fri 10/18/13</td>
<td>Fri 10/18/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Install Loose Ball Equipment</td>
<td>1 day</td>
<td>Thu 10/24/13</td>
<td>Thu 10/24/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Full Autonomous Mode</td>
<td>2 days</td>
<td>Fri 10/25/13</td>
<td>Mon 10/28/13</td>
<td>31,32,33,34</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Testing and Adjusting of Completed Robot</td>
<td>8 days</td>
<td>Tue 10/29/13</td>
<td>Thu 11/7/13</td>
<td></td>
<td>34,35</td>
</tr>
<tr>
<td>37</td>
<td>PM Writing Paper</td>
<td>47 days</td>
<td>Thu 9/5/13</td>
<td>Fri 11/8/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Project Paper Due to ATMAE</td>
<td>2 days</td>
<td>Thu 11/7/13</td>
<td>Fri 11/9/13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix G: Bill of Materials

<table>
<thead>
<tr>
<th>Team</th>
<th>Description</th>
<th>Purchaser</th>
<th>Price</th>
<th>Team Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chassis Development</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Actuators</td>
<td>Mike Cortese</td>
<td>302.29</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HD, Levers, and High steel</td>
<td>Barry Miller</td>
<td>130.12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Misc Robot Parts</td>
<td>Mike Cortese</td>
<td>26.95</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Ballbearings for Robot</td>
<td>Barry Miller</td>
<td>153.9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Motor controllers</td>
<td>Barry Miller</td>
<td>352.40</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Misc Robot Parts</td>
<td>James Mathis</td>
<td>108.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>vex Controllers, Cable and Mosser Orders</td>
<td>Barry Miller</td>
<td>452.68</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Linear Actuators (4&quot;)</td>
<td>Mike Cortese</td>
<td>238.88</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Parts for Robot</td>
<td>James Mathis</td>
<td>228.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>2004.99</td>
<td></td>
</tr>
<tr>
<td><strong>Color/ Ball Sorter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Misc Parts For Robot</td>
<td>Jon Krape</td>
<td>88.72</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6 DC gear motors</td>
<td>Gilbert Ramos</td>
<td>5247.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Misc Parts For Robot</td>
<td>Gilbert Ramos</td>
<td>1507.97</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Misc Parts For Robot</td>
<td>Jon Krape</td>
<td>250.93</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Misc Fittings, Cable Management, and Network Header Box</td>
<td>Jon Krape</td>
<td>42.07</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Motors, Data and power adaptors, Hardware</td>
<td>Jon Krape</td>
<td>401.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>2540.67</td>
<td></td>
</tr>
<tr>
<td><strong>Ball Launcher</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Misc Robot Parts</td>
<td>Kevin McNulty</td>
<td>33.77</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Basketball Hoop</td>
<td>Kevin McNulty</td>
<td>31.55</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Misc Robot Parts</td>
<td>Richard L. Stambaugh</td>
<td>26.23</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Misc Robot Parts</td>
<td>Kevin McNulty</td>
<td>9.92</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Misc Robot Parts</td>
<td>Kevin McNulty</td>
<td>141.52</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Misc Robot Parts</td>
<td>Richard L. Stambaugh</td>
<td>119.60</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Assorted Hardware for Robot</td>
<td>Richard L. Stambaugh</td>
<td>92.24</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Hardware for Robot</td>
<td>Kevin McNuty</td>
<td>62.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>523.39</td>
<td></td>
</tr>
<tr>
<td><strong>Semi Auto Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Lasers for Robot</td>
<td>Jordan Drewel</td>
<td>296.07</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Misc Parts for Robot</td>
<td>Jordan Drewel</td>
<td>212.82</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Misc Robot Parts</td>
<td>Will Funkhauser</td>
<td>261.96</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Misc Robot Parts</td>
<td>Will Funkhauser</td>
<td>15.51</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Misc Robot Parts</td>
<td>Will Funkhauser</td>
<td>16.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>904.02</td>
<td></td>
</tr>
<tr>
<td><strong>Loose Ball Recovery</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Misc parts for robot</td>
<td>Craig Tress</td>
<td>29.46</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Misc Robot parts</td>
<td>Ozgehenpierre Torres</td>
<td>15.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>45.02</td>
<td></td>
</tr>
<tr>
<td><strong>Purchase Orders</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3 color sensors</td>
<td>Bob Miller (Schneider Yasco Distrib.)</td>
<td>1863.54</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>1863.54</td>
<td></td>
</tr>
<tr>
<td><strong>MISC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Registration for the Team</td>
<td>Joel Wells</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3D Printer Material</td>
<td>Gilbert Ramos</td>
<td>857.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>887.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>GRAND TOTAL</strong></td>
<td></td>
<td>$8669.78</td>
<td>$8669.78</td>
</tr>
</tbody>
</table>
Works Cited


