iRobot Create

**2011**

Alexander (Sasha) Popov

St. Norbert College

5/3/2011



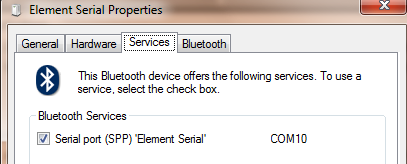
About:

* Items used in project:
* Hardware
  + Create Robot – provided by Comp. Sci. Faculty
    - www.iRobot.com
  + Optional Bluetooth dongle
    - http://store.irobot.com/product/index.jsp?productId=2803376&cp=2804606.3335976&gclid=CJG\_5YzfzKgCFYEUKgodUUoZvw&camp=Google+-+K70&s=A-UnitRank-IRBT&ab=CMS\_AccSuper\_080309&parentPage=family
* Software:
  + Visual Studio 2010
    - Free (C#) express edition: http://www.microsoft.com/express/Downloads/Project Definition

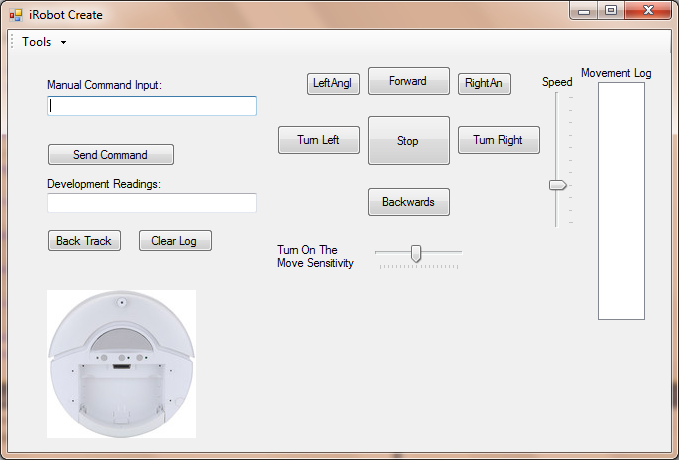
Project Definition:

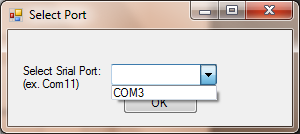
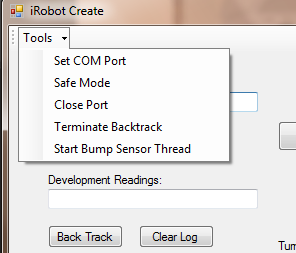
* Original Definition
  + Develop communication protocol so that two create robots can partner and cooperate activities.
* Final Definition
  + Develop a user interface to efficiently control the robot and keep track of its movement history in order to allow some autonomous programmatic control of the robot.

How To:

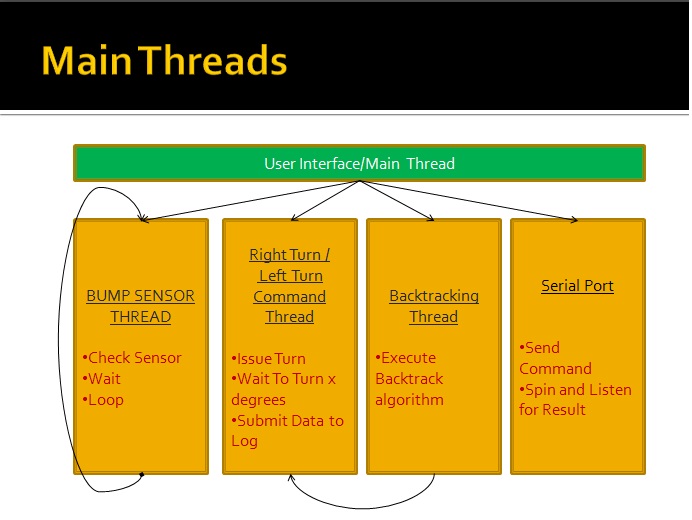
* **Hook up and Install**
  + Attach the BAM Wireless Accessory to the 20 pin I/O connector on the iRobot Create
    - http://store.irobot.com/product/index.jsp?productId=2649971&cp=2804606.3335976&gclid=CMPA3v7hzKgCFZFoKgodeFMxvQ&camp=Google+-+K70&s=A-UnitRank-IRBT&ab=CMS\_AccSuper\_080309&parentPage=family#specifications
  + Optional: Attach Bluetooth USB dongle to your computer and install driver.
    - Installation procedure depends on the operating system
  + Turn on the iRobot create by pressing the power button, ensure that the power light illuminates and is green.
  + In your operating system open Bluetooth manager and establish connection with the iRobot create.
    - Procedure depends on the operating system.
    - Default Bluetooth name of the iRobot is “Element Serial”
    - Determine the COM port of the Bluetooth connection(Note port for later use, ex. “COM10”)  
        
      * Windows 7 example
      * 
* **Compile and Link**
  + Open the project in Visual Studio 2010 and “Start Debugging” (F5).
* **Use**
  + Launch the Application(either through .exe or start debugging in VS2010)
  + Enter COM Port- you should be prompted to enter a COM port on which the iRobot create operates, if so, select the COM port from the list and press “OK”.
    - If not prompted for the port, navigate to Tools -> Set COM Port and follow the above instructions.
  + Place the iRobot into Safe Mode-after setting a COM port, navigate to Tools -> Safe Mode
    - This command places the iRobot create into “Safe Mode” as defined in the *Create Open Interface Manual(v2)* p.6
    - This function also starts the “Bump Sensor Thread”, that is, from that moment on the iRobot will stop moving once the bumper(on the front of the robot) is bumped.
    - Confirm that the lights on the iRobot are off (indicating the robot is in safe mode)
  + Drive the robot using the self explanatory UI buttons
    - Things to note:
      * if the robot is moving and the program freezes, left up the robot to terminate its movement.
    - “Turn On The Move Sensitivity” is used only with “LeftAngle/RightAngle” buttons indicating how big of an angle the robot should move while driving
      * Tap these buttons slowly allowing the robot to turn the desired angle
    - “Right/Left” buttons stop the robot and make it turn 90 degrees in respective direction.
      * You can stop the robot mid turn by pressing the Stop button
      * You cannot request two 90 degree turns at the same time, you must wait for the first turn to finish before issuing another requesting
    - You can change the speed on the move
  + Clear Log
    - Erases the log and resets the “Relative Direction” image
  + Stack
    - Visually displays the internal history stack
  + Backtrack
    - Executes the commands on the stack in the reverse and mirrored order autonomously
    - Avoid backtracking over “Turn on the move RightAngle/LeftAngle” commands as the robot has difficulty executing these commands.
    - Backtracking Exceptions
      * + Bump Sensor Thread does not work when backtracking(the robot won’t stop if the bumper is bumped as its backtracking)
        + The stop button will stop the robot but this does “hard kill” of the backtracking thread, possibly causing multiple exceptions to appear(had difficulties getting this to work)
  + General Exceptions
    - “LeftAngle/RightAngle” do not start in their own thread(freeze up UI), a feature that should be implemented in the future.
      * These functions at time freeze up the program due to racing conditions on the serial port, these functions need to be implemented in a separate thread
    - Backtracking exceptions(see above)
    - Robot turn accuracy is within +-4/5 degrees due to the delay in communication(refer to presentation’s PowerPoint diagrams for details)
    - Monitor the Robot’s battery level, as it becomes less responsive(program fails to receive info fast enough and may seem unresponsive) when the batteries approach empty levels.

**User Interface:**



Threading Overview:



* User Interface/Main Thread
  + This is the main thread which guides the program. This thread controls the user interface and depending on the user input can start other threads when needed.
* RightTurn/LeftTurn
  + This thread starts up and issues a command to the serial port to start the iRobot turning, it further idles and calculates the turn angle real time. Once the robot turns 90 degrees the thread logs the turn angle and terminates. (For more complete explanation refer to the source code).
* Backtracking Thread
  + Executes the backtracking algorithm and defined in the source code. It goes through the history stack and executes the robot’s movement history in reverse and mirrored order. It has the potential to start and terminate RightTurn/Left turn threads when needed
* Serial Port Thread
  + Serial port stays connected the whole time and is always “connected” to the robot via Bluetooth COM port. It further constantly listens for input from the robot. Also, upon request the port can send a command in bytes to the robot.