Design Topology for Applications of Low-Cost Computing Platforms to Complicated Robotic Tasks

Joseph Tran, S.G.M Hossain, and Dr. Carl Nelson
Department of Mechanical and Materials Engineering, University of Nebraska–Lincoln

Introduction

Microcontrollers

- Small computing platforms that contain a processing core, memory store, and input/output (I/O)
- Used to control various robotic tasks and perform computations

As technology advancement demanded microcontrollers with higher computing capabilities, issues of cost and design complications arose.

The current study investigates design topologies and methods of clustering microcontrollers to form low-cost microcontroller networks capable of performing tasks which are more computationally demanding.

Background

Approaches for Microcontroller Networks

- Parallel Processing – Uses shared memory among microcontrollers to gather processing power to perform appointed task.
- Distributive Processing – Tasks and computations divided among boards, data passed to each board via network communication protocols.

Arduino Fio *(Figure 1)*

- High potential as a candidate for distributive processing
- Low cost and simple to use and understand
- Several approaches for communication
  - Wired Serial (not used)
  - Wireless Serial
  - Inter-integrated circuits (I^2C)

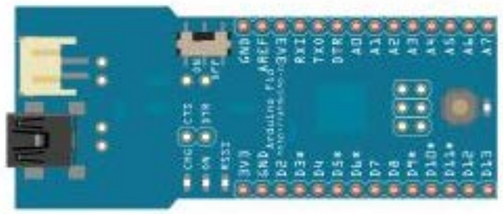

Figure 1: Open source platform, Arduino Fio

Materials & Methods

Communication Methods Tested

- Wireless Serial (*Figure 2*)
- I^2C (*Figure 3*)
- I^2C/ Wireless Serial combination (*Figure 4*)
- Each approach was tested using sensors and LED lights to observe to how the messages were passed and acted upon.

Distributive Processing Approaches

- Three-way Communication (*Figure 5*)
  - Not possible since Arduino Fio cannot perform simultaneous messaging via serial.
- Chain Messaging (*Figure 6*)

Distributive Processing Tests

- Chain Messaging (*Figure 7*)

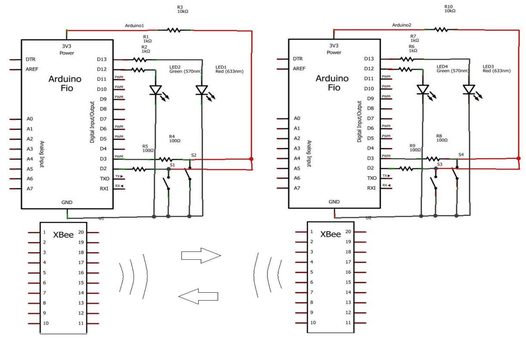

Figure 2: Wireless serial communication test. Requires the use of XBee modems to send wireless signals to other board.

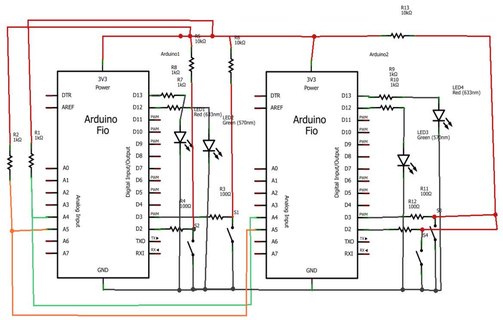

Figure 3: I^2C communication test. Uses the pins A4 and A5 on both boards to be connected for communication. Allows for communication between up to 112 devices.

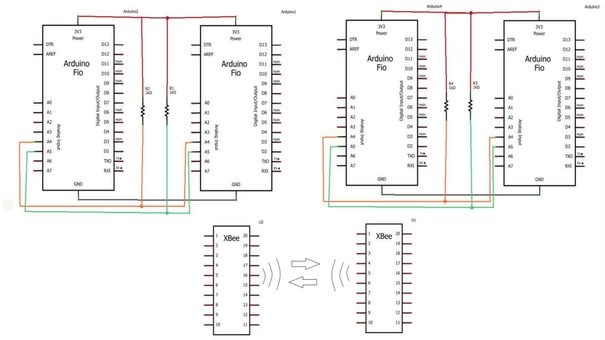

Figure 4: I^2C/ Wireless Serial combination. Uses both the wireless XBee modems and I^2C two-wire interfacing.

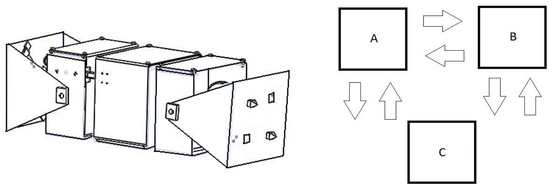

Figure 5: Modular self-reconfigurable robots (MSR) and its basic layout for three-way wireless communication.

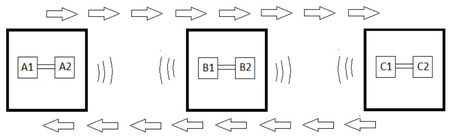

Figure 6: Basic layout for chain communication on MSR. Uses an I^2C/ Wireless serial combination.

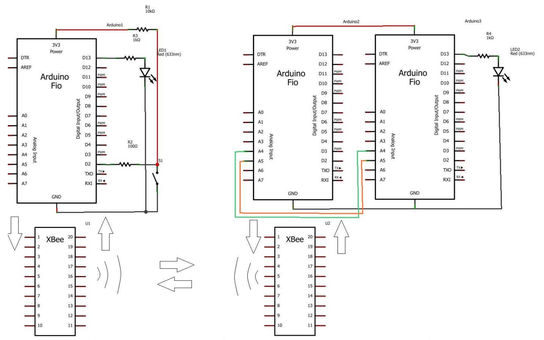

Figure 7: Chain communication uses a message passing approach. Sensor board (left) sends a signal to Xbee board (middle) which communicates with board containing LED (right) by I^2C to turn ON or OFF. The LED board then sends back a signal through Xbee board, to sensor board's LED to indicate the message has been received.

Results & Discussion

- Unable to implement three-way communication
- Capable of setting link between wireless and I^2C
- Chain messaging, using a combination between wireless serial and I^2C, is a possible approach in a distributive processing network.

Conclusions

- Higher performing microcontrollers are in need; however, this becomes a burden in cost and complexity
- Expensive microcontrollers may be replaced using a network of more affordable and simple ones
- Distributive processing network of Arduinos may be used to solve issues of cost of research, code and microcontroller complexity
- Next step is to apply topology to MSR (*Figure 5*) and explore efficiency of sharing computational tasks

Future Research

- Apply topology to MSR (*Figure 5*) and measure resulting computation times for typical tasks
- Design complete chain system for distributive processing network on Arduino
- Search for alternate modules besides XBee for wireless communication
- Work towards a complete network enabling multiple MSR modules to communicate

University of Nebraska Lincoln McNair Scholars