April 2018

Creating an Extension Board for the Khepera IV to Allow for Situated Communication

Leah Reppucci
Worcester Polytechnic Institute

Creating an Extension Board for the Khepera IV to Allow for Situated Communication

A Major Qualifying Project Report
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Robotics Engineering
By

Leah E. T. Reppucci
April 26, 2018

Professor Carlo Pinciroli, Advisor

Abstract

The purpose of this project is to design and build an extension circuit board for the Khepera IV. The circuit board allows for situated communication, which is the basis of coordinated swarm behavior, between the Khepera IV robots. This is to be done without disrupting additional extensions that can be added to the robot. The main goal of the project is to use the IR emitters and receivers on the circuit board to exchange messages between Khepera IV robots and to calculate distance and angle of the source message. The final product is two circuit boards that are created to demonstrate the desired abilities of the board and how it works with the Khepera IV. The demonstration involves the VICON and illustrates how the circuit board can work with possible interference and still send and receive IR signals.

Table of Contents

List of Figures

List of Tables

1. Introduction

The objective of this project was to create an extension board for the Khepera IV robot which would allow for an estimation of distance and inter-robot communication. This board would be used in the Worcester Polytechnic Institute NESTLab. Following the completion of this project, the board would be replicated for use with the remaining Khepera IV owned by the lab.

1.1 Objectives

- Non-disruptive to established Khepera IV extensions
- Relatively low cost per unit
- Able to communicate across the VICON field in the WPI CIBR Lab
- Send and receive communications
- Distance & angle estimation
- 360 degree communications

1.2 Schedule

This project was to be completed within two terms at WPI. The following two timelines illustrate the tentative schedule of the project and when things were to be completed. The black bars indicate when items were to be worked on, the light red highlights important tasks. The gold column indicates project presentation day at WPI, while the red column is the latest the project must be completed by.

Final Paper Schedule																	
Task	7-Jan	14-Jan	21-Jan	28-Jan	4-Feb	11-Feb	18-Feb	25-Feb	4-Mar	11-Mar	18-Mar	25-Mar	1-Apr	8-Apr	15-Apr	22-Apr	29-Apr
Introduction										Buffer	Buffer	Buffer					
Background										Buffer	Buffer	Buffer					
Methodology										Buffer	Buffer	Buffer					
Project Presentation										Buffer	Buffer	Buffer					
Results & Discussion										Buffer	Buffer	Buffer					
Conclusion										Buffer	Buffer	Buffer					
Final Paper Edits										Buffer	Buffer	Buffer					

Figure 1: Final Paper Schedule

Final Project Schedule																	
Task	7-Jan	14-Jan	21-Jan	28-Jan	4-Feb	11-Feb	18-Feb	25-Feb	4-Mar	11-Mar	18-Mar	25-Mar	1-Apr	8-Apr	15-Apr	22-Apr	29-Apr
Background Research													Buffer				Buffer
Generate Requirements													Buffer				Buffer
Initial Design Idea													Buffer				Buffer
Order Testing Materials													Buffer				Buffer
Test Sensors/Receivers													Buffer				Buffer
Choose Sensor/Receivers													Buffer				Buffer
Build Initial Boards													Buffer				Buffer
Test Initial Boards													Buffer				Buffer
Select Final Design													Buffer				Buffer
Final Board Creation													Buffer				Buffer
Test Final Board													Buffer				Buffer

Figure 2: Final Project Schedule

2. Background

2.1 Swarm Robotics

Similar to groups of insects such as ants or bees, swarm robotics utilizes a number of robots that cooperate together to perform tasks that would be difficult for a single robot [1]. The group of robots is capable of completing complex tasks is due to the robustness provided by the group, flexibility to adapt to changes, and scalability over varying group and environmental sizes [2]. The main features of swarm robotics are dispersed control throughout the group, independent actions of each individual, and simple and mostly identical members [1]. Another major aspect is the low unit cost which allows for more robots to be produced, which adds to the swarm and makes it more robust.

All robots within the swarm are required to communicate with the rest of the group, though whether this is done directly or indirectly doesn't matter. The swarm, once it begins a task, should rely only on local information which is gathered and disseminated by individuals throughout the community [5]. The biggest issues surrounding swarm robotics are that coordination and distribution algorithms are complex and must work for the entire system, which causes communications within the swarm to be unreliable. Bandwidth issues, long transmission paths, the chosen topography, and the method of communication can all interfere with the reliability and possibility of the communications [7]. The complexity of the system increases depending on the task and the number of robots. If the task is something straightforward for a single robot, such as moving an object across a room, then organizing and implementing a team of small robots to complete the same task is much more difficult due to the requirement to have the swarm synchronize and work together.

The desired topography of the swarm influences the requirements of the robots. For an interconnected topology (Figure 3.a), a large amount of communication overhead and connection handling is required for the system to properly interface and achieve the desired task(s). This means that each individual in the system must be able to handle the vast amount of information being sent and received from the rest of the swarm; requiring more allocated memory for local communication. The ring topology (Figure 3.b) greatly reduces the number of required connections at the cost of increased data transfer from node to node, similarly resulting in larger memory requirements and processing power in order to speed up the transmission of messages. The hierarchical topology is a mix of the previous two; as it tries to balance the number of connections with the amount of information being promulgated. Here, the robots at the top of the hierarchy have the most memory and processing requirements, due to the amount of information that needs to be divided and assigned to the sub-levels. These lower-levels have lower requirements due to the reduced workload per robot, an example of this can be seen in Figure 4.

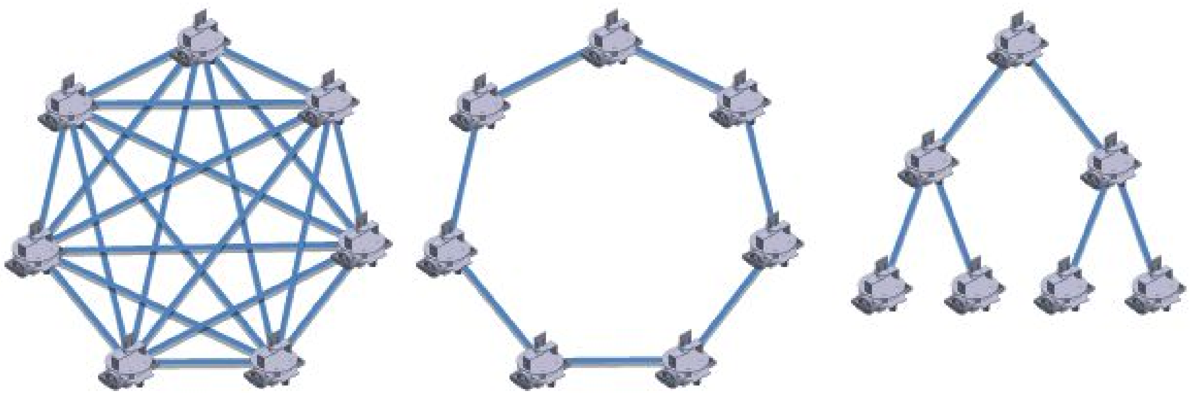


Figure 3: Situated Communication Topology

a. Interconnected Topology; b. Ring Topology; c. Hierarchical Topology [3]

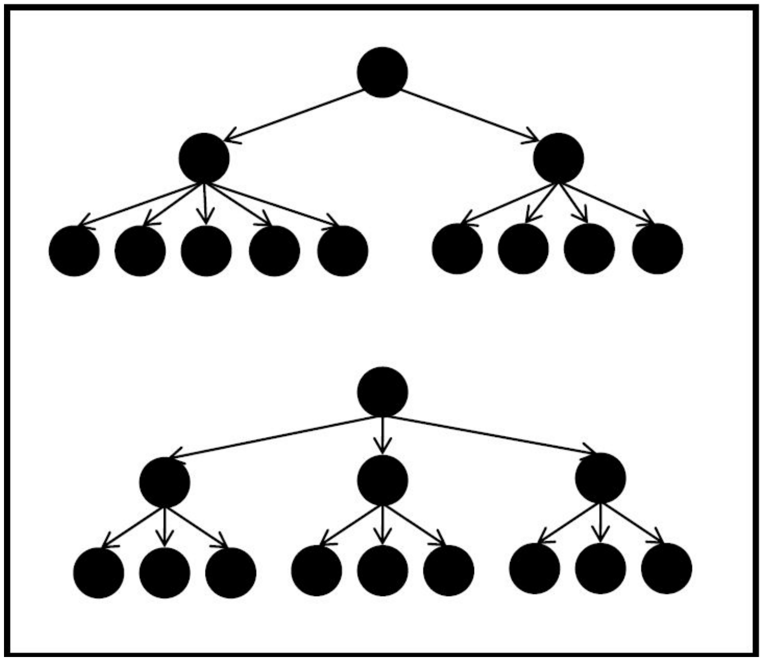


Figure 4: Hierarchical Topography

The structure before the addition of another mid level node (top), the structure after the addition of another mid level node (bottom). Notice that the workload of all mid level nodes has been reduced as a consequence. [3]

2.2 Situated Communication

Communication for multi-robot systems can take one of two forms. The first, abstract communication, is widely used due to the low cost and increased efficiency [8]. Abstract communication is when the method of transporting a signal does not matter, only the message holds value. In comparison, situated communication relies on both the method of communication as well as the message itself to convey meaning [8]. An example would be identical data transferred using IR and bluetooth. In situated communications, the final interpretation of the data would be different for the IR verses the bluetooth message since the method of communication varied.

Abstract communication is dependent on the robot knowing not only its own location but the ability to integrate information received from the rest of the group into its world map. This is where the strength of the system can fail, since if the world map or the localization is incorrect, then the system will fail. Something as simple as a moving object or unplanned shifting/moving of the robot can completely disrupt the map and cause the localization to fail. Although it is possible to recover from these disturbances, the amount of time required is high.

Situated communications allows for actions to be taken based on sensor input directly. The biggest issue is deciding on the best method to distribute and store the data throughout the swarm. Since human intervention is not possible when dealing with swarms, the communications network is prone to topology changes as random nodes fail [12]. Redundancy in the form of replication of data across numerous nodes increases reliability and ability to recover data in the event of failed nodes [9]. Redundancy brings complications since as robots fail, and leave and enter the system the storage must be redistributed in order to maintain the reliability [9]. Achieving redistribution requires large data transfers throughout the system, which monopolizes bandwidth and postpones the transmission of new information.

2.3 Budget

The total budget for this project is $500. This includes cost of materials for the final circuit boards as well as the prototypes and any unexpected costs.

3. Methodology

Using the background information and research, the following steps were created in order to complete the required objectives. This involved initial testing of parts, the creation of a prototype, testing of the prototype, and finally creation of the final project. Prior to anything, first a tentative design was created to help shape initial testing.

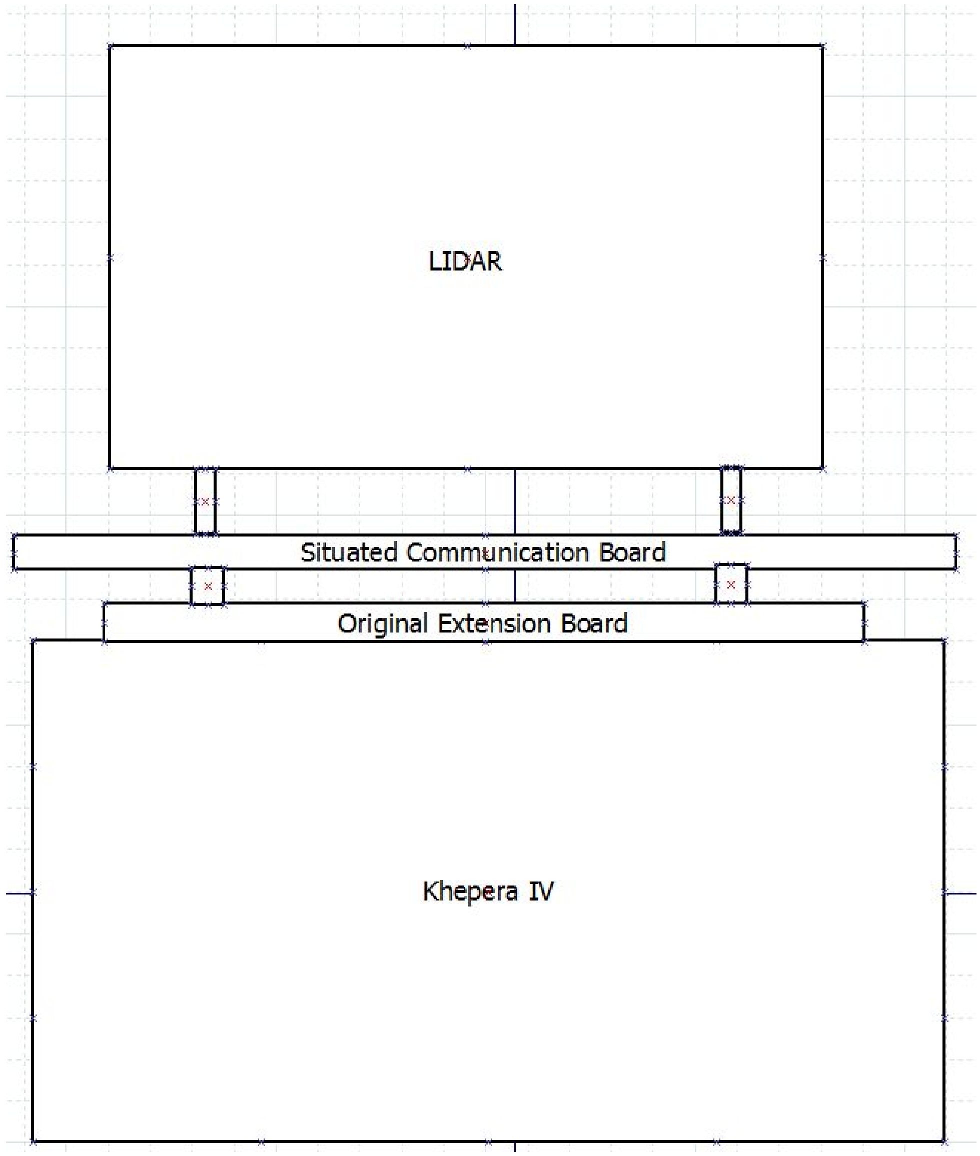


Figure 5: Initial Design Overview

3.1 Initial Part Testing

Research was required to decide which components could be considered for use in the final printed circuit board and that need to be ordered for initial testing. This involved combing through digikey and other electronics suppliers to compare potential emitters, receivers, and transceivers. 5 components were ordered based on the following criteria: range, angle, cost, and required current and voltage. The required current and voltage for the components were not the priority when looking for parts due to these factors being controllable by external elements. A list of the purchased parts from the initial order can be found in Appendix B.

The testing occurred once all the parts had arrived. Testing was done using two Arduino Uno's as the microcontroller with two external breadboards and a variety of external resistors. Each part was tested according to the following criteria: angle, distance, resistor(s), and cost.

The testing was done using two Arduino Uno and the Arduino IDE. A library was downloaded from GitHub called Arduino_IRemote by user Z3t0, which allowed for easier programming of the IR sensors in the IDE. The wavelength to be emitted was altered by changing the ir_Sony.cpp code to match the desired wavelength. Appendix E shows an example of alterations made to IRsendDemo. These alterations allowed for checking of the signal accuracy since the emitted signal was known and the receiver's serial monitor was checked for the expected message.

The parts were also tested at the VICON field to ensure that they would not be influenced by the 850 nm IR that the VICON uses. The datasheets of each of the parts were checked to ensure that they would work at wavelengths other than 850 nm, the tests mentioned above were carried out again to guarantee successful use.

The initial selection of parts resulted in the selection of the TSOP38238 IR receiver for the prototype. The two emitters purchased during the first order worked, but did not meet the criteria to make it to the prototype. This resulted in a second purchase order focused solely on emitters. These emitters were tested the same way as those from the first part order and the Grove Infrared Emitter was ultimately selected for the prototype.

3.2 Prototype Board

Based off the results of the initial testing combined with the end goals for the circuit board, duplicate parts were ordered for the prototype. Additionally new parts had to be researched and purchased to allow for the successful connection between the prototype and the Khepera IV. These chosen parts can be found in Appendix C. The final prototypes were soldered and tested in a week. The soldering followed the design shown in Figure 6.

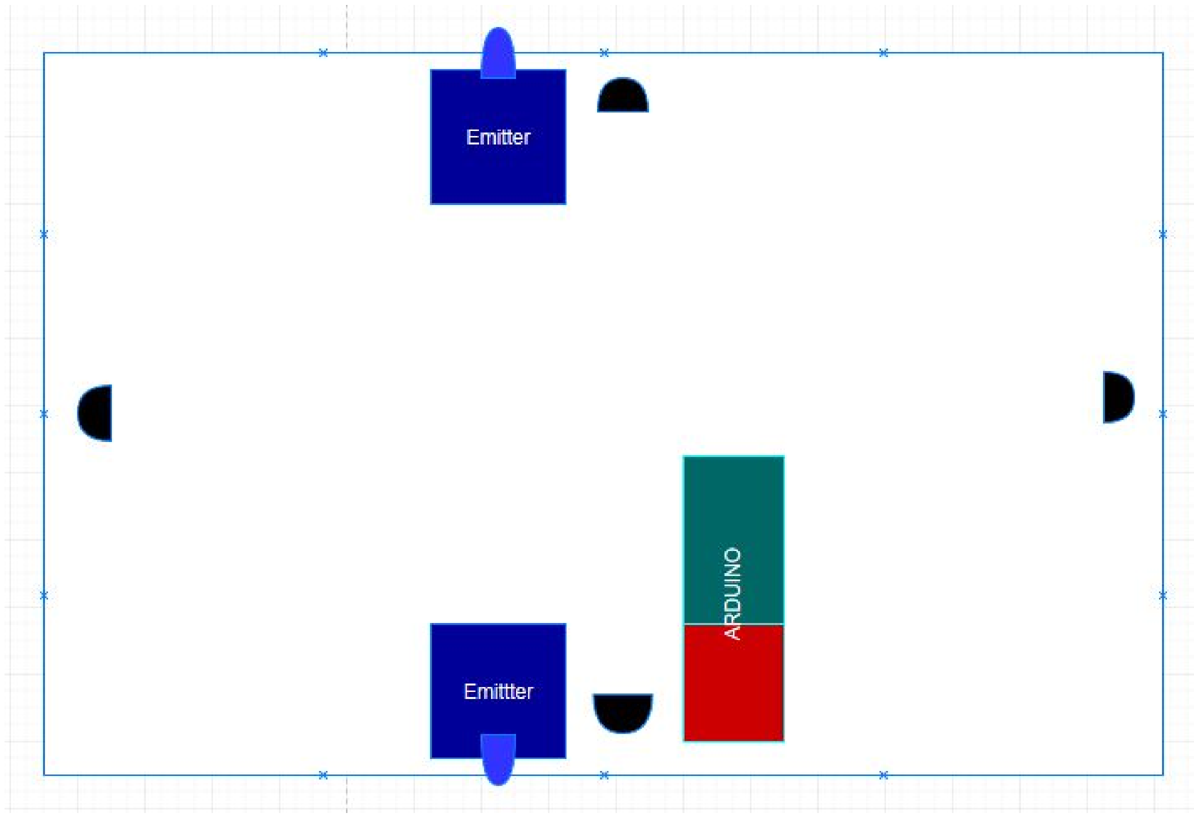


Figure 6: Updated Board Design

Prior to the arrival of all the parts, a tentative schematic was planned using ExpressSCH , so that the proper placement and connections between each part were known and troubleshot prior to the assembly of the board. This can be seen in Appendix D.

Testing was carried out similarly to that of the initial part testing except that now tests were done to check for proper solder and to check that the design was feasible. Research and calculations were completed to ascertain the best way to find the distance and angle of an incoming source message.

3.3 Final Prototype Board

The results of testing the prototype caused a revision in the initial design which had relied on the use of the pins from the Khepera IV for programming and power. Following the revision, it was decided to purchase a microprocessor which could be connected via a cord to the Khepera IV. Parts were ordered and then soldered to create two final prototype boards for use in proof-of-concept for the design and calculations. This also allowed for testing of the capabilities of the receiver and emitter, to check if they were capable of receiving and interpreting the necessary information to allow for the distance and angle calculations.

4. Results and Discussion

4.1 Initial Part Testing

In order to test the individual parts, several criteria, both qualitative and quantitative,were chosen to help narrow down potential parts for use in the prototype.

Table 1: Initial Part Testing

Part	Distance (m)	Angle in ° (From Centerline)	Setup Requirement(s)	Coding Requirement(s)	Unit Cost ($)
UV Emitter	0.5	45	Resistors	Adjust frequency	14.36
IR Point Source Emitter	0.5	45	Resistors	Adjust frequency	11.13
TSOP IR Receiver	45	70	Capacitor, resistor	N/A	1.12
Photodiode	1	65	Resistor	Convert binary into message	3.08
Phototransistor	1	80	Resistor	Convert binary into message	3.72

The UV emitter worked well. It was possible to extend the range of the UV emitter using capacitors and transistors as well as through increasing the current and reducing the duty cycle. The drawback with this was the amount of space needed to extend the range, especially when considering the space required multiple emitters and extenders on the prototype. Coupled with the high cost per unit, ultimately the UV emitter was deemed impractical.

This high unit cost also made the IR point source emitter impractical. This cost issue was once again compounded by the lack of space available for multiple emitters and extenders on the prototype.

The TSOP IR receiver stood out immediately upon use. Despite the more complex setup required for use, the part had a wide receiving angle which when combined with the ability to decode the message across long distances made this part ideal. This contrasts sharply with the photodiode and phototransistor. These parts were chosen to test in order to show why they wouldn't work for the final circuit board; both the photodiode and phototransistor acted like a switch when exposed to the IR emissions. Although this didn't mean that these parts were unusable for this project, it did mean that additional coding would be required in order to convert the series of signals into data that could be used by the receiving robot. These parts also required a filter since they would activate due to a range of frequencies between visible and IR light. Since the IR receiver would only respond to a set frequency, the receiver required less additional coding.

The UV and IR emitters purchased in the first part order did not meet the requirements which led to a second part order which focused on emitters. Two emitters were singled out long distance transmissions, the Grove IR Emitter and the Open-Smart IR Transmitter Module. The Grove IR emitter was chosen for the prototype since could send data over more than 3 meters and had a wide and strong signal. This signal strength was especially apparent during testing since even with objects such as boxes and humans in between the emitter and receiver, the signal was still received with minimal errors. The shipping speed and low cost of the Grove emitter also helped in this decision. The Open-Smart IR mitter, despite paying for 1 week shipping, took almost two weeks to arrive. Due to the time constraints of this project, that length of time automatically made the Open-Smart emitter unviable.

4.2 Prototype Building and Testing

During the initial building of the prototype, it was recognized that the initial design for the board would not be feasible due to the difficulties in procuring connectors to and from the Khepera

IV. The connectors used by the Khepera IV are very small and require roughly two weeks of wait time since they need to be manufactured and shipped. The use of the connectors also complicates the programming since available pins are limited which results in careful examination of the K-Team schematics prior to programming the new board and increases the chances of mistakes being made. The prototype included the TSOP Receiver and the Grove IR Emitter which were soldered onto the EFRobot protoboard. This was then tested to ensure that signals could be sent and received from all angles. During the initial testing, the number of receivers on the board was increased from two to four. The rationale behind this was that with only two receivers, the angle of the incoming signal was difficult to determine since the signal could come from anywhere within ~180 degrees per-receiver. The increase to four receivers allowed for a reduction in possible location of the incoming-signal since two receivers could interpret the same signal to better ascertain the source.

It was initially determined that the Inverse Square Law should be used for the distance estimation. This was because the irradiance of the emitter could be calculated. It was unknown though if the receiver was capable of determining the W/m^2 of the incoming signal.

$$\frac{Intensity_1}{Intensity_2} = \frac{Distance_2^2}{Distance_1^2}$$

From this equation, the distance of the source of the incoming signal can be determined. This is because $Intensity_2$ is known from calculations done using the datasheet for the IR Emitter. $Intensity_1$ is calculated by the receiver as it interprets the incoming message. $Distance_2$ is 0 m since the light hasn't traveled anywhere when the intensity is being calculated at the source while $Distance_1$ is the unknown that is desired to be discovered. It was later realized that the Inverse Square Law could not be applied since with the initial distance being 0 m, the resulting calculations would always be either zero or undefined. Priority was then given to finding a new way to calculate relative distance.

The Law of Cosines was then used to calculate the angle the incoming signal was from in regards to the front of the receiving robot, which is 0°. Figure

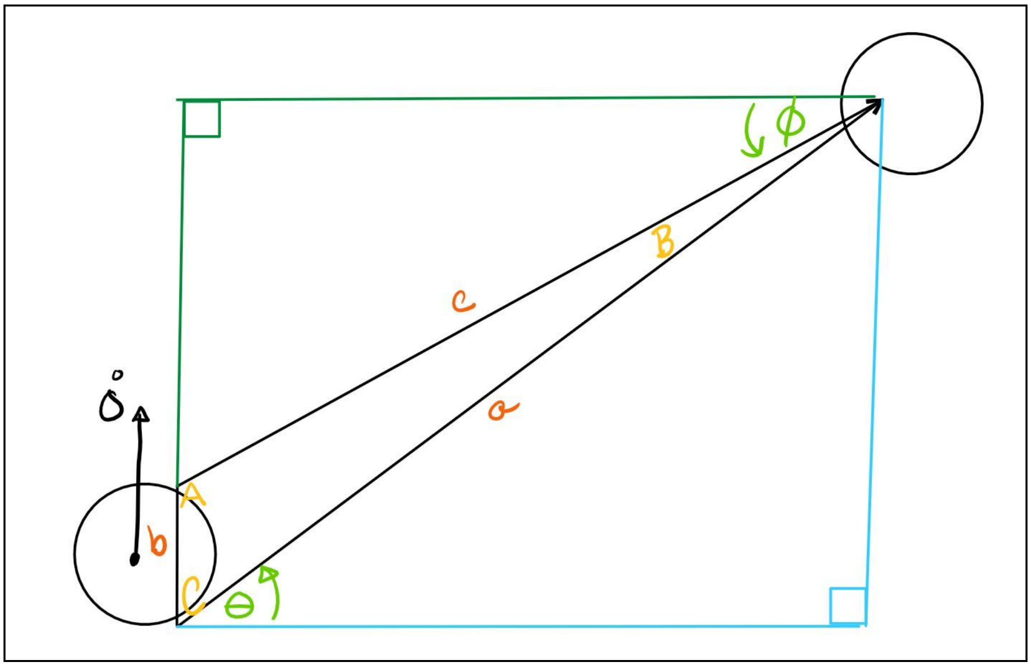


Figure 7: Law of Cosines Diagram

$$c^2 = a^2 + b^2 - 2ab\,cos(C)$$

$$cos(C) = \#$$

$$cos^{-1}(cos(C)) = cos^{-1}(\#)$$

$$C = cos^{-1}(\#)$$

$$90 - C = \Theta$$

This math can be repeated for angle A, which then allows for the solving of Φ.

4.3 Final Prototype

Due to an overestimation in the amount of time it would take to code, the ability of the receiver to interpret an incoming signal to allow for distance and angle estimation was not tested with the initial prototype board. Therefore, the final prototype board had to not only demonstrate a potential design but also the ability to estimate the incoming signal origin.

The distance estimation was completed using the analog-in data from the receiver. The numbers, ranging from 0 to 1023, were averaged at varying distances, graphed, and a trendline calculated, which allowed for approximate distance estimation to be achieved. The code used to gather data to create this trendline can be found in Appendix F. The biggest problem with this approach to distance estimation was that the sensitivity of the sensors varied. For example, on one protoboard, the TSOP connected to port A0 resulted in numbers ranging between 1000 and 1023, while the receiver in port A3 gave results around 340 ± 5 depending on the distance from the emitter. This meant that every receiver had to be calibrated in order to allow for code reuse for distance calculations.

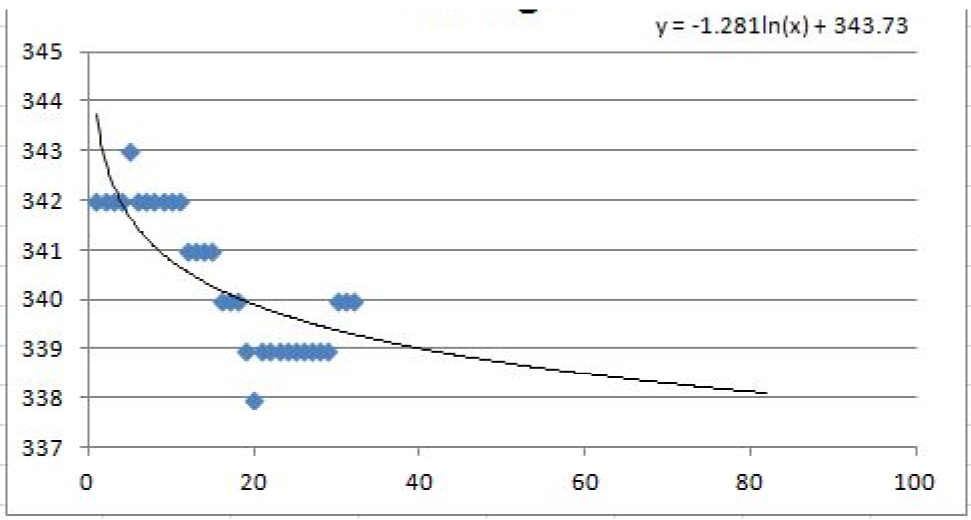


Figure 8: Distance (in) to Analog-In Trendline

As seen in Figure 8 above, despite the increasing distance from the source, the change in analog-in was very small which meant that distance estimates were not overly precise; and the estimate would become less reliable the further the source was from the receiver.

Conclusion

The goal of this project was to create an extension board for the Khepera IV which would allow for situated communication. The following objectives were created in response to this goal in order to determine the success of the project:

- Non-disruptive to established Khepera IV extensions
- Relatively low cost per unit
- Able to communicate across the VICON field in the WPI CYBR Lab
- Send and receive communications
- Distance & angle estimation
- 360 degree communications

All of these goals were met at some level. The board works without disrupting the other extensions available for the Khepera IV while keeping the cost per board below $50, which includes shipping and handling which was a large expense. When purchasing the IR emitter and receiver, it was ensured that the IR frequencies were not the same as the VICON's frequency. Achieving communications, especially 360 degree communications, was the simplest and most straightforward objective to reach since communications was the basis of this project and required only the proper parts positioned equally around the board. The distance and angle estimation was the only objective that wasn't quite reached due to the use of the trendlines which caused a variability between individual receivers as well as in the accuracy of the estimates. Future work on this project should involve designing, ordering, and testing PCBs. These boards should include eight TSOP receivers to allow for the best angle estimation. Testing should be conducted using all the Khepera IV available in the lab, with each robot assigned its own individual frequency to allow for identification and to check that the board can handle the amount of signals. Finally, further exploration should be done to either increase the accuracy and precision or to find and implement a new method for the distance estimate.

Bibliography

[1] I. Navarro and F. Matía, "An Introduction to Swarm Robotics", *ISRN Robotics*, vol. 2013, pp. 1-10, 2013.

[2] M. Dorigo and E. Şahin, "Guest Editorial", *Autonomous Robots*, vol. 17, no. 23, pp. 111-113, 2004.

[3] N. Alunni, R. Goloski, A. Haggerty and E. Jones, *Hierarchical Swarm Robotics*. Worcester Polytechnic Institute, 2011.

[4] M. Brambilla, E. Ferrante, M. Birattari and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective", *Swarm Intelligence*, vol. 7, no. 1, pp. 1-41, 2013.

[5] A. Sharkey, "Swarm robotics and minimalism", *Connection Science*, vol. 19, no. 3, pp. 245-260, 2007.

[6] Y. Tan, "Swarm Robotics: Collective Behavior Inspired by Nature", *Journal of Computer Science & Systems Biology*, vol. 06, no. 06, 2013

[7] T. Rodrigues, M. Duarte, M. Figueiró, V. Costa, S. Oliveira and A. Christensen, "Overcoming Limited Onboard Sensing in Swarm Robotics Through Local Communication", *Transactions on Computational Collective Intelligence XX*, pp. 201-223, 2015.

[8] K. Støy, "Using Situated Communication in Distributed Autonomous Mobile Robotics", in *SCAI'01 -Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence*, Odense, Denmark, 2001.

[9] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright and K. Ramchandran, "Network Coding for Distributed Storage Systems", *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539-4551, 2010.

[10] L. Keller, E. Atsan, K. Argyraki and C. Fragouli, "SenseCode", *ACM Transactions on Sensor Networks*, vol. 9, no. 2, pp. 1-20, 2013.

[11] "Tunable Sparse Network Coding for Multicast Networks", in *2014 International Symposium on Network Coding (NetCod)*, Aalborg, Denmark, 2014.

[12] A. Bagchi, P. Vyavahare, I. Altaf Gillani and T. Gupta, *Decentralized Random Walk-Based Data Collection in Networks*. 2017.

Appendix A

	Date Ordered	Item	Quantity	Cost Per Item ($)	Total with Shipping ($)
	1/31	UV Emitter	1	14.36	
	1/31	IR Point Source Emitter	1	11.13	
	1/31	TSOP38238 IR Receiver	1	1.12	
	1/31	Photo Diode	1	3.08	
	1/31	Photo Transistor	1	3.72	
	2/24	Sullins Connector Solutions NPPC252KFMS-RC	2	4.83	
	2/24	EFRobot FIT0193	1	3.15	
	2/24	Sullins Connector Solutions SBH11-NDBC-D25-SM-BK	2	1.93	
	3/8	Grove - IR Emitter	9	3.9	
	3/8	TSOP38238 IR Reciever	9	1.12	
	4/8	Arduino Pro Mini 328 - 5V/16MHz	2	9.95	
	4/8	Sparkfun Serial Basic Breakout-CH340G	2	6.95	
		Grove - IR Emitter	1	3.9	
		OPEN-SMART Infrared Emitter Module IR Transmitter Module for Arduino	1	2.85	
		Con Header 50POS 1 MM DL AU T/H	1	9.22	
		1MM Micro STRIPS	1	6.58	
	4/8	EFRobot FIT0193	3	3.24	
Running Total				154.85	240.78

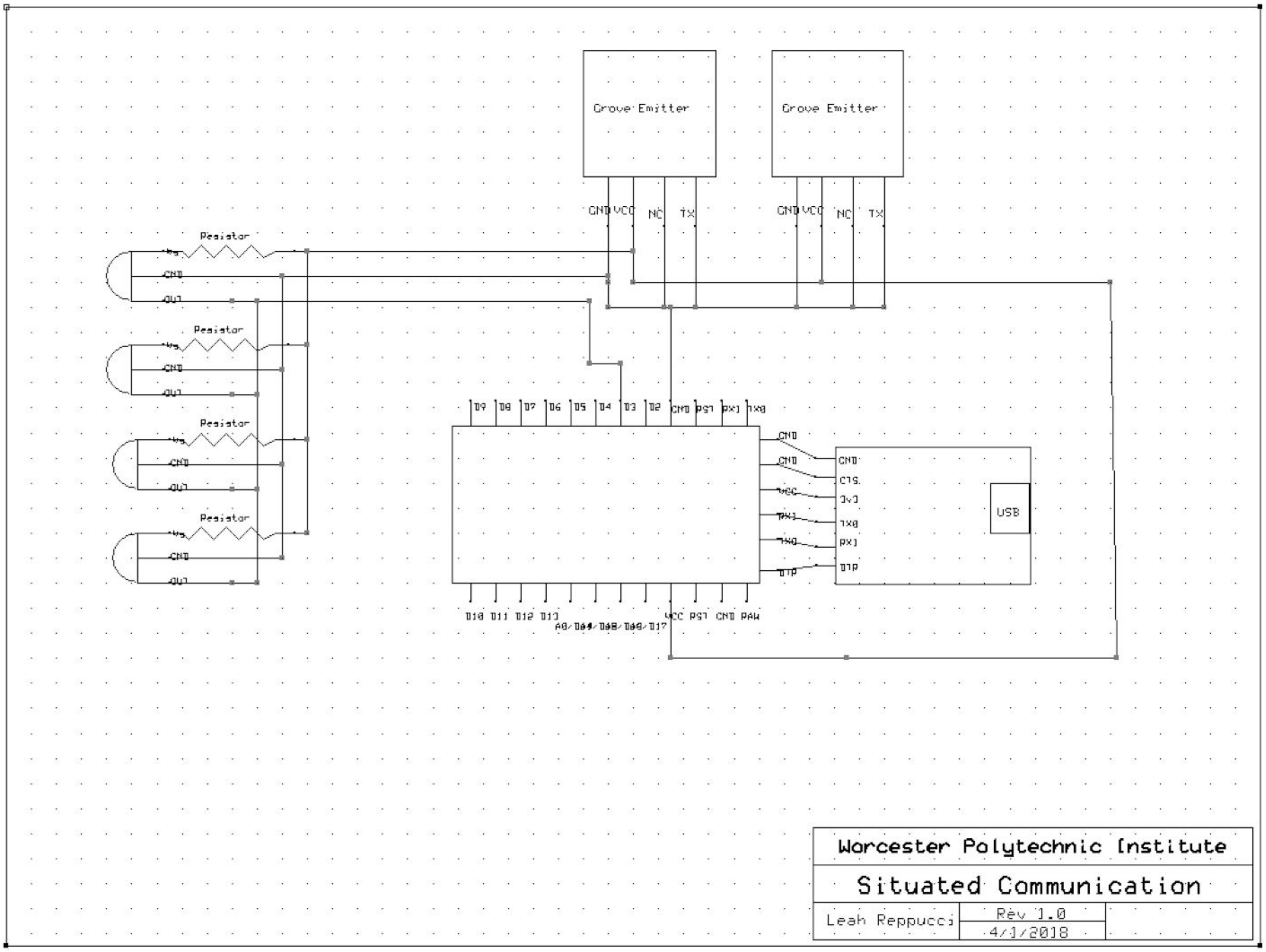
Appendix B

Part	Link	Quantity	Approximate Cost	Total Cost	
UV Emitter	https://www.digik	1	14.36	14.36	
IR Point Source Emitter	https://www.digik	1	11.13	11.13	
TSOP38238 IR Receiver	https://www.digik	1	1.12	1.12	
Photo Diode	https://www.digik	1	3.08	3.08	
Photo Transistor	https://www.digik	1	3.72	3.72	
					Total w/ Shipping
					42.4

Appendix C

Part	Link	Quantity	Approximate Cost	Total Cost	
Grove - IR Emitter	https://www.seee	9	3.9	35.1	
TSOP38238 IR Reciever	https://www.digik	9	1.12	10.08	
					Total Cost w/ Shipping
					76.21

Appendix D


Grove Emitter
Grove Emitter
GND VCC NC TX
GND VCC NC TX
Resistor
Resistor
Resistor
Resistor
USB
Worcester Polytechnic Institute
Situated Communication
Leah Reppucci
Rev 1.0
4/1/2018

Appendix E

IRsendDemo §

```
/*
 * IRremote: IRsendDemo - demonstrates sending IR codes with IRsend
 * An IR LED must be connected to Arduino PWM pin 3.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 */


#include <IRremote.h>

IRsend irsend;

void setup()
{
}

void loop() {
  for (int i = 0; i < 3; i++) {
    irsend.sendSony(i, 12);
    delay(40);
  }
  delay(5000); //5 second delay between each signal burst
}
```

Appendix F

Distance_Calculation

```
int sensorRead = A2; //Readings from sensor sent to A0
float x;
const int numberReadings = 10;
int readings[numberReadings]; //readings from analog-in
int readIndex = 0; //index of the current reading
int total = 0; //running total
int average = 0; //average


void setup() {
  Serial.begin(9600);

  for (int thisReading = 0; thisReading < numberReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}

void loop() {

  //read from the sensor
  readings[readIndex] = analogRead(sensorRead);
  //add the reading to the total
    total = total + readings[readIndex];
     //advance to next position in array
    readIndex = readIndex + 1;

  //check if at end of array
  if (readIndex >= numberReadings) {
    //wrap around to the beginning

    //calculate average
    average = total / numberReadings;

    String lead = "The average is ";
    String tog = lead + average;
    Serial.println(tog);
    delay(500);
    average = 0;
    total = 0;
    readIndex =0;
  }
  delay(200);
}
```