Robust Object Recognition for Mobile Sensor Nets

David C. Lee
Networked & Embedded Systems Lab
University of California, Los Angeles
1762 Boelter Hall
Los Angeles, CA 90025
davidlee@ee.ucla.edu

Jonathan Friedman
Networked & Embedded Systems Lab
University of California, Los Angeles
1762 Boelter Hall
Los Angeles, CA 90025
jf8@ucla.edu

ABSTRACT

In this paper, we describe a method for providing object recognition to mobile sensor nodes using RFIDs. Because mobile sensor platforms tend to have very noisy electrical environments, the mobile platforms with RFIDs must be carefully designed so that the noise does not affect the ability for the reader to sense an RFID tag. We created an application that uses the RFID reader on the Ragobot moving in a random walk that searches for tags and reads and writes to them.

Categories and Subject Descriptors

B.0 [**General Hardware**]

General Terms

Design, Verification

Keywords

Controlled Mobility, Robots, RFID.

1. INTRODUCTION

Mobility has become an important research topic in sensor networks. One application of mobility is in robogaming, where sensor nodes must collaborate to achieve a final goal while navigating a terrain. The Ragobot is a sensor network platform designed with robogaming in mind. With its small-form factor and dual-motor traction platform, it is able to traverse complex terrains.

One aspect of sensing in robogaming that has not been explored is object recognition. While an IR sensor can be used to detect the presence of an object, the robot has no ability to determine what that object is. RFIDs provide a cost efficient solution to object recognition. RFIDs can be used to tag all the objects on the game board and small RFID readers mounted on the Ragobot could read the tags and determine the object type based on the information stored on the tag. There is very little data processing required since it is mostly done by the reader itself.

The contributions of this paper are threefold. First, we argue the case for the widespread deployment of an emerging sensing modality for local object detection and classification. Second, we perform an extensive analog analysis of an emerging hardware platform for mobile distributed sensor networks – proving its suitability for the field. Finally, the authors implement an example deployment in which an equipped robot explores a target space identifying appropriately pre-classified objects with near absolute perfect accuracy.

Several sensing modalities exist for performing object detection and classification. The most common in robotics is infrared (IR). Current work in IR has emphasized either low hardware complexity at the cost of low resolution or high resolution at the cost of intense mathematical complexity. The high-end systems perform high-resolution image processing tasks on a dense IR detector array while the low-end systems (designed mostly for simple binary collision avoidance) are comprised of only 2D sparsely populated arrays.

Many high-end approaches, like that of [1], have focused on rigorous profiling of the detector's behavior in combination with advanced statistical models. In [1] the authors employ a self-adapting extended Kalman filter to an infrared image to filter atmospheric jitter and thus derive the true sensed reading for extrapolation of the object's classification and presence. Their results were impressive as they improved accuracy seven-fold when compared with performing the ranging analysis on the raw image, but the process itself is computationally cumbersome and probably beyond the realm of practicality for today's battery constrained robotic embedded systems. A promising low-end alternative ([3]) involves using a rotating emitter to scan the target horizontally and compare it with a set of known profiles. In effect, the approach first classifies the target as one of several known "types", then fuses this type information with the received IR profile and detector sensitivity profile to determine range and angle. It was highly effective (97%) at classification and surprisingly good (for a single emitter system) at precise target detection. But [3] is an active system requiring both a rotating scan head and continuous emission of IR. In our intended application we seek to minimize the impact on battery consumption as well as conserve IR bandwidth. Unlike [3], our proposed approach must be able to scale to multiple Ragobots operating in the same general vicinity.

Excepting IR, another common approach to the classification task involves the use of visible light camera systems to process images and perform classification in a manner similar to the human brain. The authors of [4] propose using such a camera to detect the human eye in an image and replace it with an artificially rendered eye. Such a scheme would solve the common video teleconferencing problem where the remote person looks at the screen, instead of the camera. In doing so, the remote person's image does not make eye contact with the television audience. However inspiring their results, such schemes are severely processor intensive and well beyond the capabilities of a mobile sensor node in terms of energy and weight (many approaches, like [5], call for multiple cameras operating simultaneously).

2. RAGOBOT

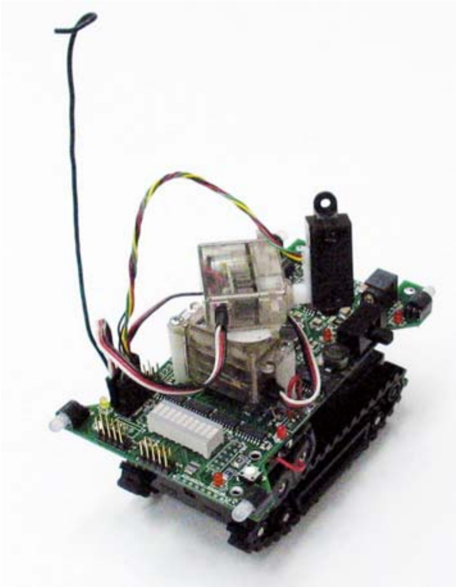


Figure 1. The Ragobot

The Ragobot is an emerging mobile sensor network platform targeted at lab-scale environments (see fig. 1). The robots, themselves, are small (approx. 14cm x 8cm top-down surface area), lightweight (under 1kg including battery), and high torque (60 deg vertical climb) allowing them near unlimited mobility in the face of obstructed environments. We began with an unverified, single, partially instrumented robot and proceeded to build additional sensor modules, annunciation components, battery assemblies, battery chargers, and an entire second robot.

RFID, unfortunately, is extremely sensitive to high frequency noise injected through the power supply and over the air. The small transmitters in the tags respond to emitter noise by frequency shift, which, if appropriately severe, can drive the tag's response transmission completely out-of-band. In the more mild cases, read range is drastically reduced. It, therefore, becomes essential that the host platform provide an extremely AC and HF quiet operating environment for the RFID host module.

This specific result is an example of a larger dilemma facing mobile sensing systems. Because of the energy constraints imposed by the battery's limitations and the application's desire for the longest possible lifetime, the use of a high-efficiency power supply is essential. However, these systems, along with the actuation components (motors, servos, etc) generate a substantial amount of noise – both in ground wave propagation (EMF) and magnetic radiation (EMR). Noise levels may not only disrupt sensing, but communications as well. It is absolutely essential that the analog performance of any proposed mobile sensing platform be effectively characterized such that demonstrable performance can be confidently accepted as typical.

In this regard, we began by analyzing the sources of noise injection on the ground plane of the Ragobot. The Ragobot hosts a Berkeley Mica2 mote with fundamental oscillator frequencies at 32.768kHz and 7.3728MHz. The main power supply on Ragobot comes in just under a fundamental 90kHz (89.98kHz). The battery charging regulator (Maxim's MAX1908) switches at a nominal 400kHz, but was not evaluated due to assembly limitations preventing its installation on the robot. In addition to the regular sources, the presence of digital data transmissions, whose edges represent extremely high frequency noise, complicates the analysis. Other, traditional, noise problems can occur including: VHF/UHF RF reception on parallel traces, Hall effect cross-talk (inductive coupling), HF cross-talk (capacitive coupling), and current loop EMR.

The end effect of all of these sources is a large spread-spectrum noise-nightmare waiting to assault our sensing and communications sub-systems. To combat the threat, a plethora of techniques were applied in the design stages of the Ragobot. While these efforts represent previous work by the authors, their description and evaluation presented here does not. We begin with a theoretical discussion of the mechanisms by which noise is attenuated. The majority of EMR and EMF concerns originate from the Biot-Savart law (eq.1). Here we present a simplified form making the assumption of a centerline mounted RFID reader which experiences worst-case conditions from the Ragobot circuit board overhead. α is the plus/minus angle to the highest current trace oriented perpendicular to the RFID reader's read direction (500mA). ρ is the mean distance over the arc defined by (+ α , - α) between the reader and this aggressor trace [6].

$$H = \frac{\frac{-1}{2} I \sin(\alpha)}{\pi \rho} \quad \text{(eq.1)}$$

For the current revision Ragobot, α is about 30 degrees and ρ is 5.673 centimeters. This yields H = 1.403 *I*, where *I* is the worst case current condition. The specification for worst case motor current (the highest single source) is 500mA, so we double it for safety and demonstration and use *I* = 1A. H = 1.403 Henry. Applying the Ampere Circuital Law (eq. 2) we see that coupling through the air at such close range injects a current at half the value of incidence – 500mA in this example!

$$I = 2 H \pi \rho \quad \text{(eq. 2)}$$

It is particularly important to note that this exercise has made the assumption that the current return path imprinted on the circuit board is in far-field to the aggressor trace when compared with the RFID circuit. In reality this was specifically made not so. Ohm's Law guarantees that current follows the shortest return path (least R), so our design specifically counters noise injection by minimizing the lengths of paths. Minimizing these lengths reduces α . Separating the aggressor trace from the RFID reader increases ρ. Under the logic of equation 1, the Ragobot design routes the highest current motor drive paths along the far opposite side of the robot where ρ is effectively maximum.

In carrying the rebuttal to our initial far-field assumption further, we arrive at an additional tactic employed in Ragobot's design –

planing. Close to the underbelly of the robot, inside the FR4 plastic material which comprises its decking lies a virtually solid metal sheet. In our Ampere model we can consider the aggressor trace as a surface of sheet current K where K is normalized by equation 3, in which b is the width of the trace (0.012") and *I* is the total current in the trace (again, we'll assert 1A as an exaggerated value) [6].

$$K = \frac{I}{b} \quad \text{(eq. 3)}$$

Because we are forcing such an exorbitant amount of current through such a narrow trace, the normalization gives us K = 3030A/m (meter of width)! Solving Ampere over a uniform charge sheet of infinite length (we'll relax this assumption in a moment), gives equation 4 [6].

$$H = \frac{1}{2} K \, x \, An \quad \text{(eq. 4)}$$

Here, *An* is a normalized directional vector between the point of analysis and the sheet current, *K*. Even though this magnetic field acts through a small angle of incidence to the reader due to the cross-product, the effect is still quite significant. Now we add the influence of the ground plane, which is a second parallel sheet of current of equal magnitude but opposite sign. Equation 4 is now twice its magnitude in the inter-plane region, but *H* = 0 outside the planes as the fields cancel! [7] Even if we now relax the infinite length assertion on the aggressor trace, the ground plane is still approximately infinite in length by comparison validating our earlier assumption.

The previously discussed approaches involving layout, material choices, and routing attempt to mitigate the generation of spurious interfering signals. Our second line of defense involves those techniques targeted at mitigating the propagation of noise should it, invariably, be generated. The crux of these techniques is first and second order analog filtering. RC filters have a low pass frequency response if the signal is inserted through the resistor and output across the capacitor [7]. In all cases of integrated circuit power connection to the Ragobot circuit board, use is made of the characteristic resistance of the circuit trace in tandem with a low-ESR tantalum capacitor. The parts were specifically picked to provide about 50dB rejection at 1MHz and roll off at 10dB per decade (the best performance the authors could find without using extremely expensive ultra-low ESR parts). These distributed 1st order filters form the core of the filtering strategy. For several especially critical power sections, such as the feed to the communications sub-system, a second order filter system is installed. Analyzing the performance of the analog filtering was done initially by hand and then, subsequently, by Cadence software's pSpice package in a previous work by the authors. The Cadence model appears in figure 2 and the frequency-domain results appear in figure 3.

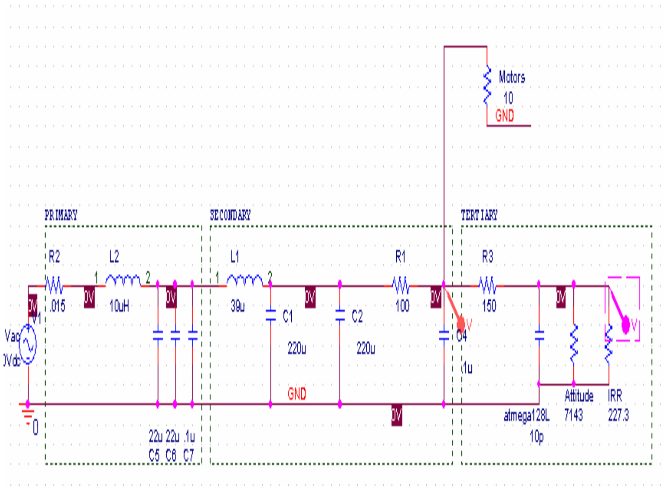


Figure 2. Cadence model of a second order filter system

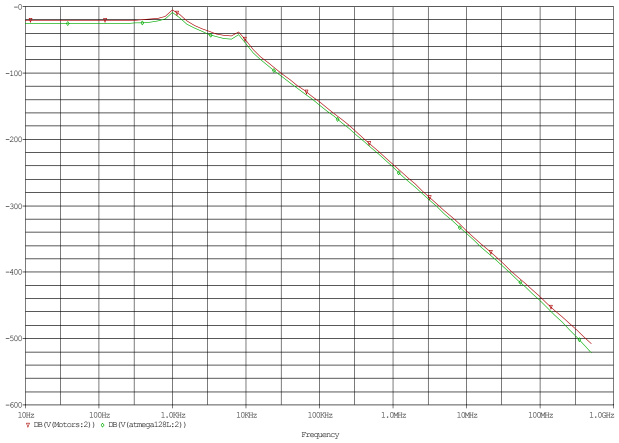
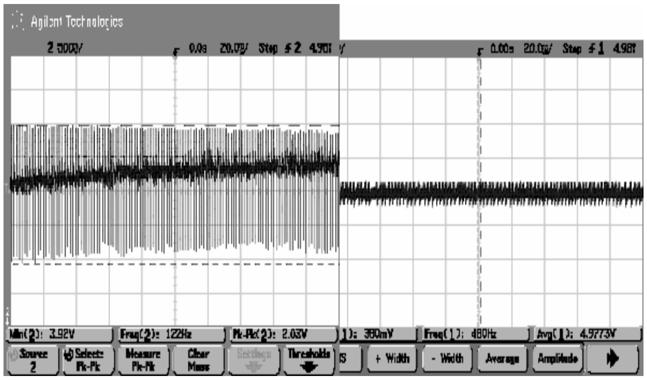


Figure 3. Frequency domain results of a second order filter system

The simulations left the verification question open however. In this current work, the authors proceed to close the loop and evaluate the actual performance in the hardware for comparison with the prediction. Measuring the robot under full motor load (the worst case) produced figure 4. The predicted broadband suppression was 19dB. The observed value was only 7.277dB, but this difference rapidly disappears, and by 1MHz the actual performance of the hardware is past 200dB suppression (the limit of our analyzer)in perfect accord with our prediction.



Agilent Technologies

2 500mV/ 0.0s 20.0?/ Stop 2 4.98?

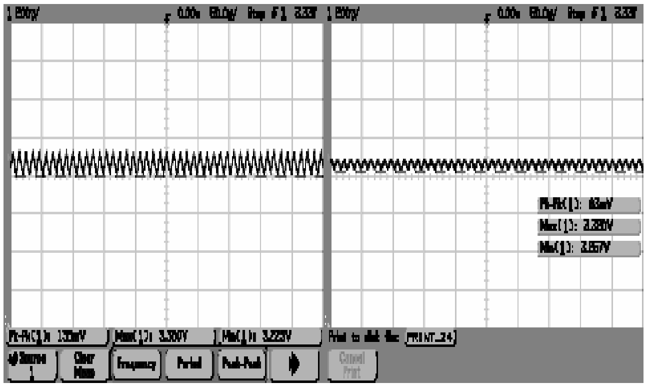
0.00s 20.0?/ Stop 1 4.98?

Min(2): 3.92V Freq(2): 122Hz Pk-Pk(2): 2.03V

(1): 380mV Freq(1): 480Hz Avg(1): 4.9773V

Figure 4. Noise Rejection on 5V power supply rail

The second-order filter produces the following results which match our simulated predictions with surprising accuracy. The third stage filter network has a predicted broadband response of –3dB. As is shown in figure 5, the measured broadband response was -2.648 dB. Again, we see high frequency suppression in excess of 200dB above 1MHz as expected. The aggregation of the noise rejection techniques employed in Ragobot is remarkably capable – able to hold the final load regulation error to within just 0.9% in the face of a 2V swing under a ½ Ampere loading condition with almost 95% efficiency.



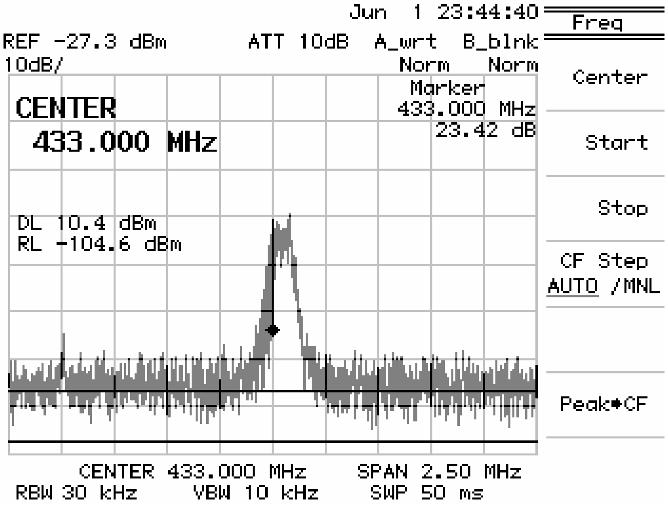
Pk-Pk(1): 133mV Max(1): 3.300V Min(1): 3.223V

Pk-Pk(1): 63mV

Print to disk file: PRINT_24

Figure 5. Tertiary (final) filter stage output

The final analog analysis performed involves intentional RF emissions from the Ragobot platform and verifies the operating environment for the communications sub-system. For reference we compare with a free standing Mica2 mote. The two figures (6 and 7), taken on the same scale, confirm that the noise floor remains unchanged and that the radio still operates in band and without drift when attached to the Ragobot. Additionally, using the ground plane of the Ragobot as an isotropic reflector, the Mica2 radio experiences an additional 9.7dB of gain for the same constant power output level.



Jun 1 23:44:40

REF -27.3 dBm ATT 10dB A_wrt B_blnk

10dB/ Norm Norm

CENTER 433.000 MHz

Marker 433.000 MHz 23.42 dB

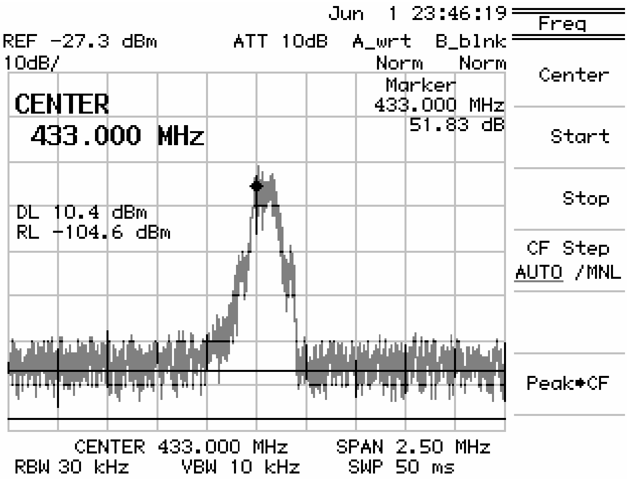
DL 10.4 dBm

RL -104.6 dBm

CENTER 433.000 MHz SPAN 2.50 MHz

RBW 30 kHz VBW 10 kHz SWP 50 ms

Figure 6. Radio output from a Mica2 Mote



Jun 1 23:46:19

REF -27.3 dBm ATT 10dB A_wrt B_blnk

10dB/ Norm Norm

CENTER 433.000 MHz

Marker 433.000 MHz 51.83 dB

DL 10.4 dBm

RL -104.6 dBm

CENTER 433.000 MHz SPAN 2.50 MHz

RBW 30 kHz VBW 10 kHz SWP 50 ms

Figure 7. Radio output from the Ragobot

3. RFID

3.1 Hardware

For the RFID reader, we chose to use the Skyetek M1 RFID reader. It operates at 13.56 MHz and supports many different protocols such as the ISO 15693 [2]. Because of its small form factor of 38 mm x 40 mm (see fig. 8), it was easier to mount on the Ragobot than any other reader, like the Texas Instruments Series 2000 micro reader. Another benefit of the Skyetek M1 is that it does not consume much power. According to the datasheet, it consumes about 45mW using a 5V supply, but from our measurements, the reader uses between 30-35 mW using the 5V supply on the Ragobot.

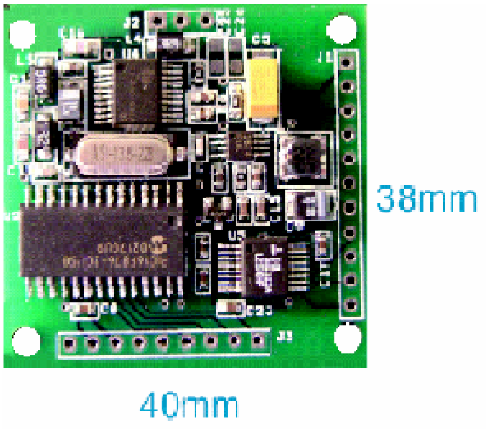


Figure 8. Skyetek M1 RFID reader [2]

Another benefit is that it has a built-in antenna, or one can mount another antenna to the reader. From our measurements of the radio range using different sized tags, the internal antenna has a vertical read range of 7 cm above the tag and a write range of 5 cm. If the antenna is 0.5 cm above the tag, then the horizontal read and write range is about 1 cm away from the edge of the reader. If the reader is raised higher, then part of the tag needs to be directly under the reader for it to be detected.

The RFID reader can communicate with a microcontroller using RS232, TTL, I^2C, or SPI. We used the TTL serial input to communicate with the Mica2 mote because of the simple serial interface and SPI does not function on the Mica2 because the designers of the Mica2 hacked it for the radio. Because the voltage supply in the RFID reader is 5V and the voltage supply in the Mica2 is 3.3V, we built a voltage level shifter circuit using a diode and the internal pull-up resistors of the Atmega128L microprocessor on the Mica2. Otherwise, the 5V voltage level on from the transmitter of the RFID reader could damage the serial receiver of the Mica2, rendering the serial port useless.

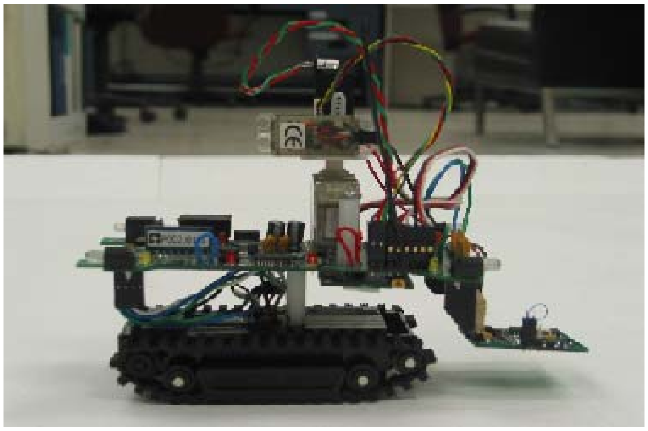


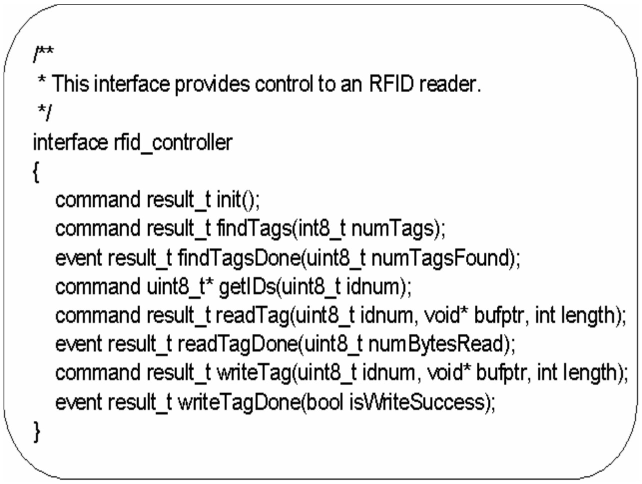
Figure 9. RFID mounted on the Ragobot

We mounted the RFID reader to the front of the Ragobot, where the gripper used to be (see fig. 9). This allows the robot to perform object recognition for the tagged objects in front of it. Because a long read range would cause the reader to detect all the tags within range, we chose the short range internal antenna so that the Ragobot can only detect and recognize objects close to it.

From our measurements, at a 9600 bps, the latency between sending out a command and receiving the response is about 47 ms (21ms to send the 16-byte read command packet, 16 ms until the response is sent, and 10 ms to receive the 12-byte response packet). Also, with the internal antenna, the reader can only consistently read 2 tags simultaneously.

3.2 TinyOS Driver

The Ragobot currently runs on TinyOS so the driver for the RFID reader is also written in nesC for TinyOS. It provides applications with an interface to find, read, and write to tags (see fig. 10).



```
/**
 * This interface provides control to an RFID reader.
 */
interface rfid_controller
{
   command result_t init();
   command result_t findTags(int8_t numTags);
   event result_t findTagsDone(uint8_t numTagsFound);
   command uint8_t* getIDs(uint8_t idnum);
   command result_t readTag(uint8_t idnum, void* bufptr, int length);
   event result_t readTagDone(uint8_t numBytesRead);
   command result_t writeTag(uint8_t idnum, void* bufptr, int length);
   event result_t writeTagDone(bool isWriteSuccess);
}
```

Figure 10. TinyOS RFID reader driver interface

The findTags command tells the RFID reader to search for a *numTags* number of tags. If *numtags* equal 1, then the driver sets the RFID reader to a fast mode that only searches and returns the first tag read. Otherwise, the driver sets the reader to the slower "inventory mode" where multiple tags are returned. When all the driver reads all the tag IDs, it stores them into an array. A signal is sent to the application stating that *numTagsFound* tags were found. The user can access the IDs by calling getIDs with an index into the ID table.

The readTag command allows the application to read a *length* number of bytes from a specific tag with the index *idNum* into a buffer *bufptr*. This means that findTags must be called first to find all the tags in the vicinity before a read can begin. It signals to the application that the data is available with the number of bytes read from the tag.

Similarly, the writeTag command allows the application to write a *length* number of bytes from a buffer *bufptr* into a specific tag with the index *idNum*. It signals to the application if the write is successful with the Boolean isWriteSuccess.

Internally, the driver maintains its own state, allowing only one command to be processed at a time. If the driver is busy fulfilling a previous command, then any subsequent command sent returns a FAIL. Otherwise, it returns SUCCESSFUL and tries to fulfil the command. The reader can still return with a response that says that it cannot read/write/find tags so numTags, numTagsFound, and isWriteSuccess are important in determining if the command was truly successful.

4. APPLICATION

4.1 Random RFID Search Application

In order to test our new system, we created an application in which the robot drives around on a game board. If it detects an RFID tag, it would read it and then write some information back to the tag. The robot would then randomly decide on which way to turn and continue driving until the next tag is found. Potentially the tags could contain localization information as well as treasure and hints about the environment.

For this application, we implemented a simple state machine as shown in fig. 11. The robot starts in the MoveFind state, in which the robot moves around on the board looking for RFID tags. Once a tag has been detected, the robot is told to stop its motors and try to read from the tag, and it enters the StopRead state. If a recently discovered tag cannot be read from, then the robot assumes that it has driven too far, and the robot should backup until the tag can be read from—state Backup. Once the tag is successfully read, the robot randomly determines which direction to turn (state DetermineDirection). Then it writes this information onto the tag in state WriteTag. Once the write is finished, the robot turns in the direction determined earlier and finally the robot moves and tries to find more tags.

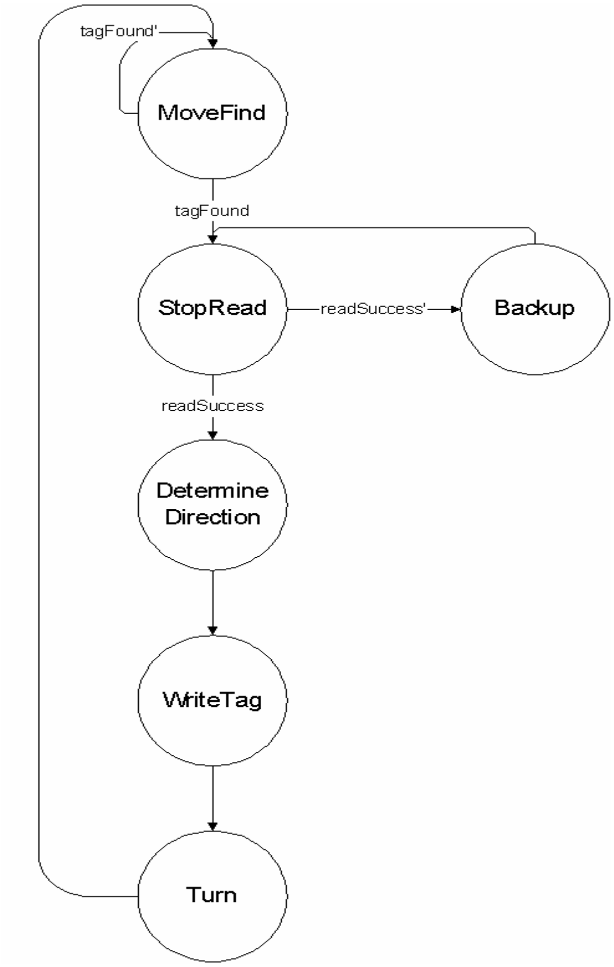


Figure 11. State machine for application

This application worked well. The Ragobot would drive over the tag and stop immediately. It would then proceed to align itself to the tag if it could not get a proper reading from the tag. It would proceed to write to the tag and all the lights would turn on to announce that the write was successful. Then the Ragobot would drive forward, turn left, or turn right depending on the random function and continue to drive until it detected another tag.

4.2 Addition of Obstacle Detection

We added obstacle detection using the IR ranger on the Ragobot to the application. This would have prevented the Ragobot from running into any obstacles while searching for RFIDs. Unfortunately, the Mica2 mote was not able to handle running the IR ranger, the RFID reader, and the motor control simultaneously. The problem is that all three drivers block interrupts for a certain period of time. With all three drivers running simultaneously, there were too many hardware and software interrupts. Many of the interrupts were missed because all the drivers turned off software interrupts. So while the IR ranger was able to function with the motor controller, the RFID driver was unable to read all of the incoming packets from the serial port. Most of the time, the Ragobot would drive over the tags without detecting them.

In order to get multiple drivers working simultaneously, the timing model for the driver end needs to be reformulated and reevaluated. The drivers need to be faster and cannot spend all of their time blocking interrupts. Otherwise, the robot will be unable to respond to all of its sensors.

5. CONCLUSION

We have presented the case for the widespread applicability of an emerging sensing modality for local object detection and classification. Namely, we have proposed using passive RFID tagging devices coupled with a mobile reader unit to navigate among various environmental objects while simultaneously classifying them by unique or group ID's. We further demonstrated the utility of our scheme by deploying new Ragobot hardware into a field exercise in which it successfully traversed a bed of RFID tags. In demonstrating tag reads and writes, it confirms the analysis performed. The Ragobot is a sufficiently quiet operating environment conducive to mobile RFID operation.

The inability to use many sensors simultaneously is a software timing issue with the drivers. The drivers need to be written such that interrupts do not get blocked. Otherwise, the system will not be able to handle all the interrupts from the sensors and the motor controller.

6. REFERENCES

[1] P. Maybeck, D. Mercier, "A Target Tracker Using Spatially Distributed Infrared Measurements."

[2] Skyetek M1 RFID Reader Datasheet

[3] T. Aytac, B. Barshan, "Differentiation and Localization of Target Primitives Using Infrared Sensors."

[4] Weiner, D. Kiryati, N., "Virtual Gaze Redirection In Face Images." *Proceedings of the 12th International Conference on Image Analysis and Processing, 2003.*

[5] *Junior, B.M.; de Oliveira Anido, R.;* Motion and Video Computing, 2002. Proceedings. Workshop on, 5-6 Dec. 2002 Pages:187 – 192

[6] W. Hayt Jr., J. Buck. *Engineering Electromagnetics: Sixth Edition*.

[7] A. Kahng, S. Muddu. “An Analytical Delay Model for RLC Interconnects”

[8] Skyetek Protocol v2.0 Host Communication Description

[9] Application Note 002: Using the Skytek Protocol: RFID Tag Commands