SECE Location Tracking

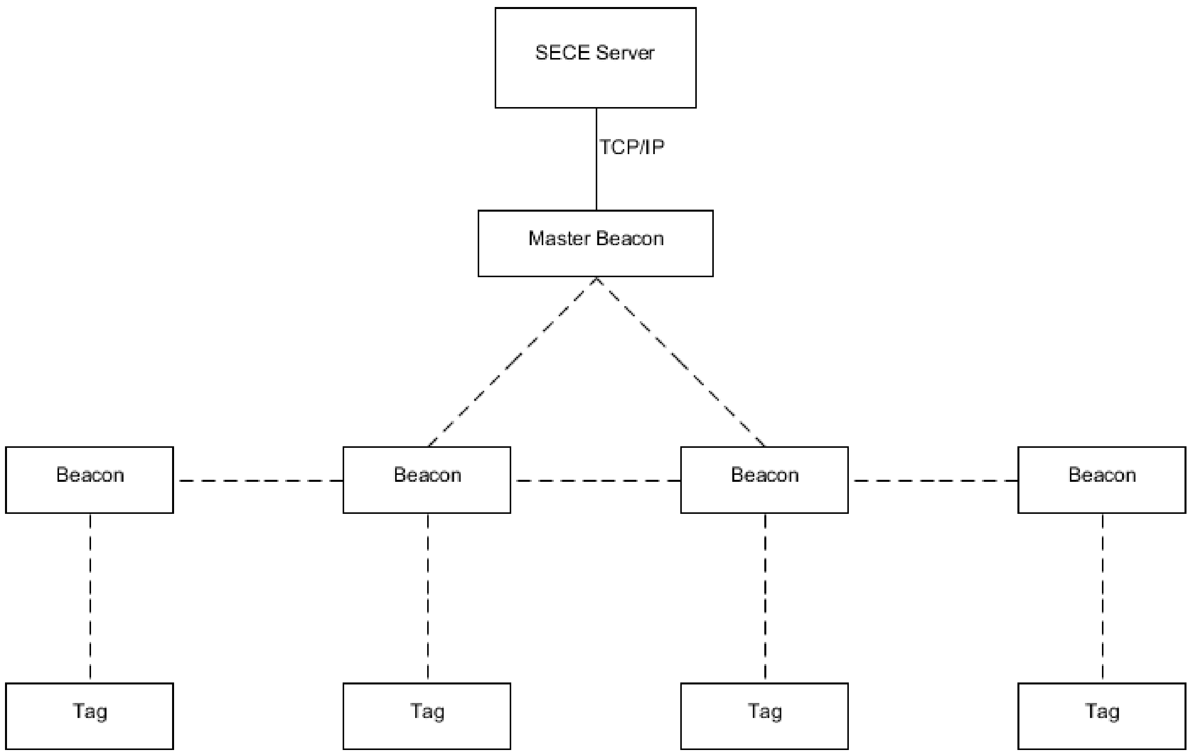
Zachary Salzbank

Fall 2011

1 General Description

The location tracking system in SECE uses wireless communication to determine the location of an object. The system comprises of two types of components: beacons and tags. Beacons are stationary objects that transmit a signal wirelessly at a predetermined interval of time. Tags are placed on the object that will be tracked. When the beacon signal is received by a tag, the distance to the beacon is determined. The tag then transmits the identifier of the nearest beacon back to the system.

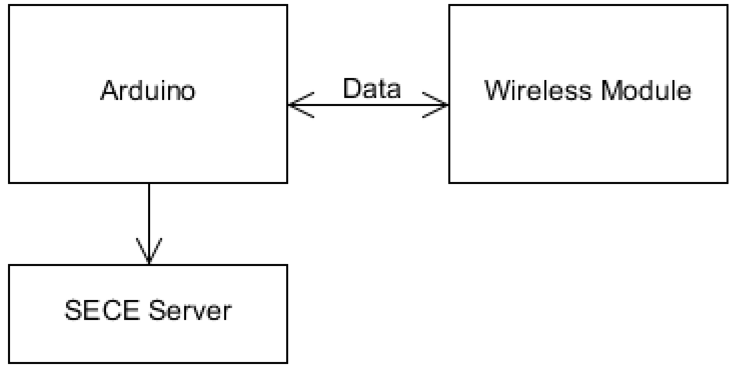
2 System Diagram



3 System Components

3.1 Master Beacon

The master beacon is an Arduino based device that has the ability to communicate with the SECE server through an Ethernet connection. In addition, it communicates wirelessly with other beacons and tags. Only one master beacon may be used at a time.



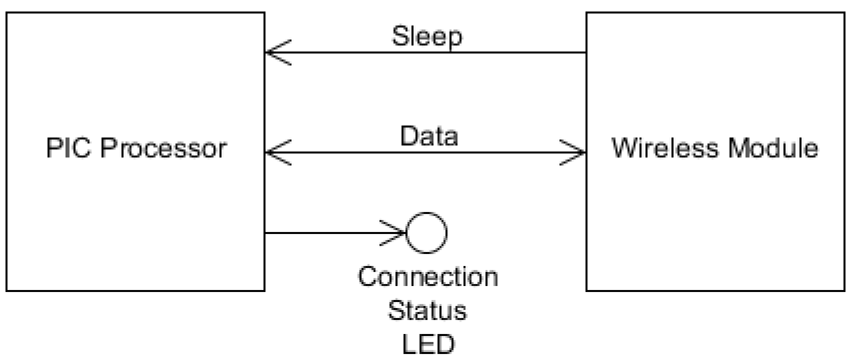
The master beacon performs three functions:

- Acts as the sleep coordinator for wireless communication (see Power Conservation below),
- Receives messages sent by tags and relays them to the SECE server, and
- Performs the functions of a standard beacon.

Messages that are received from tags contain the identifier of the tag that transmitted the message and the identifier of the beacon that is closest to the tag. The master beacon relays this data to the SECE server.

3.2 Beacon

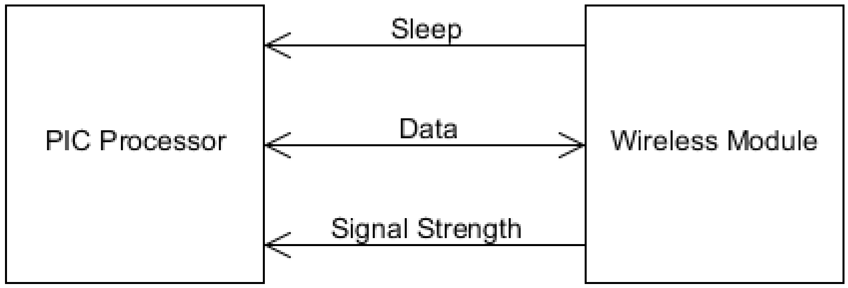
A beacon is a PIC processor based device that communicates wirelessly with the tags. Multiple beacons may be used at the same time. However, there should not be more than one beacon in a room.



Beacons transmit a message at a predetermined interval of time that is broadcast to all tags within range. In addition, beacons indicate whether or not they can communicate with the network via the Connection Status LED.

3.3 Tag

A Tag is a PIC processor based device that communicates wirelessly with the beacons. Multiple tags may be used at the same time.



Tags receive messages from the beacons. Upon reception, the tag determines the distance to the beacon. If the distance of beacon is closer than the previous distance to a beacon, the tag will send a message wirelessly to the master beacon. The distance to the closest beacon is updated each time a message is received from it. However, if no message is received for a certain amount of time, the beacon will not be considered closest anymore.

4 Details of Operation

4.1 Wireless Communication

Wireless communication is achieved through the use of XBee DigiMesh modules. These components are configured to create a mesh network. The mesh network allows each module to act as a message router. This means that the beacons do not have to be within range of the master beacon, as long as a path exists to the master beacon through intermediate beacons. XBee modules were chosen instead of the standard ZigBee modules because they allow easier creation of mesh networks and synchronized sleeping.

Each module has a unique ID that is used to identify it to the rest of the system.

4.2 Power Conservation

The beacons and tags conserve power by sleeping at a predetermined interval. This interval is set on the XBee modules. The module on the master beacon acts as the sleep coordinator. It allows the other modules to sync with the master beacon, so that all modules sleep and wake at the same time. In addition, the PIC processors will go into a sleep mode at the same time as the XBee modules.

Beacons will broadcast their message during the awake period of the sleep cycle. When tags wake from sleep, they first determine if a location update needs to be sent to the master beacon. If it does, the message is sent. Then the tag listens for any new beacon messages and processes them.

4.3 Message Formats

All messages use the API mode of the XBee protocol to communicate. The data listed below is the application specific data that is sent in the message.

4.3.1 Broadcast Message

The broadcast message is sent from the beacons to all other devices within range. It will not be broadcast further than one hop away; it cannot use any other module as an intermediate path. Other beacons will ignore the broadcast message. Only tags will process this message.

The broadcast message is one byte long.

Byte	0
Field Description	Header
Value	0x00

4.3.2 Closest Message

The closest message is sent from a tag, directly to the master beacon. It will use as many intermediate beacons as necessary to reach the master beacon.

The closest message is nine bytes long. The data sent is a header, followed by the ID of the XBee module of the closest beacon.

Byte	0	1	2	3	4	5	6	7	8
Field Description	Header	Closest Beacon ID							
Value	0x01	XBee Module Dependent							

4.4 Determining Closest Beacon

The distance to a beacon is determined by measuring the signal strength (RSSI) of the message that was sent from the beacon. Higher signal strength means the beacon is closer. Signal strength is obtained from the XBee module.

Too many beacons in an enclosed area will make this measurement inaccurate. For best results, the tag should be 2-4 feet away from a beacon as well. Beacons are best placed on a ceiling.

Initially, it was thought that the RSSI value would not be accurate enough to track tags. An IR light was going to be placed on each beacon, so that a sensor on the tags could determine if they were in the same room together. However, after lowering the transmission power on the XBee modules, this was determined to be unneeded and just the RSSI value is used.

4.5 Beacon Connection Status

A beacon is determined to be connected to the network if it receives broadcast messages from other beacons. If a beacon is not connected to the network, the connection status LED will illuminate. The beacon checks to make sure it receives a message during every wake period. If it goes too many periods without receiving a message, the beacon is not connected to the network.

5 Results

5.1 System Operation

The overall system works as planned. Tags can be tracked when they come in range of a beacon. When multiple beacons are in the same room but separated by a distance, the tracking works too.

During development, there were some issues that were eventually resolved:

- Beacons were relaying broadcast messages through the entire network, making tags seem like they were close to all beacons.
- Multiple beacons being in the same room were causing reliability issues.
- RSSI value readings were inaccurate initially due to the way they were being obtained.

5.2 Power Consumption

Initial power calculations were done to determine how long the battery of a tag would last on a CR2303 (watch) battery without and power conservation features (i.e. sleeping).

Part Name	**Average Current Draw (mA)**
XBee DigiMesh Module	50
Processor	0.007500033
Variables	**Value**
Sleep Period (s)	1
Awake Time (s)	1
XBee Max Amps Awake (mA)	50
XBee Max Amps Sleep (mA)	0.05
Battery mAh	225
Processor Max Amps Awake (mA)	0.00015
Processor Max Amps Sleep (mA)	0.00000065
Current Adjustment	1.1
XBee Data Rate (kbps)	250
TX/RX Data Bytes (bytes)	200
Calculations	
Awake Times Per Hour	3600
Percent Awake	1
Percent Sleep	0
XBee Average Current Draw (mA)	50
Processor Average Current Draw (mA)	0.007500033
Required Awake Time	0.0064
Results	
Battery Life (h)	4.090295544
Battery Life (d)	0.170428981

These calculations proved to be fairly accurate. Upon measurement of the current draw of an actual tag, the average was approximately *0.0574* amps over the period of an hour. This resulted in a calculated battery life of *3.920* hours, a 4% difference from the initial estimate.

When power conservation features were added (a sleep period of 5 seconds and an awake time of 1 second), the average current was measured to be *0.015* amps over an hour. This resulted in a gain of about 5 times the battery life to approximately *19.6* hours.

5.3 Future Work

Additional work can be done to make the system even better:

5.3.1 Reduce RSSI Fluctuation

The RSSI value is currently read each wake period and recorded. A better way to do this would be to average the RSSI value over a period of time. This would reduce fluctuations in the RSSI value, which would allow more accurate calculation of location when multiple broadcast messages are received by a tag.

5.3.2 Reporting Battery Status

Beacons could report battery status to the master beacon when the available power is becoming low. This would assist in determining why a tag is not reported in a location that it is known to be in.

5.3.3 Tag Panic Button

A button could be added to tags that will send out a special message immediately when the next wake cycle begins. This could be used as a panic button to alert the system of some kind of condition.

5.3.4 Detecting when a Tag is Out of Range

Currently, the beacons have no way of knowing when a tag is out of range. Only the tags can determine this state. If the tags kept track of when they went out of range, they could report this time to the beacons when they come in range again. This would allow the system to keep track of periods when it is unsure if the location of a tag is accurate or not.

5.3.5 Reduce Lost Messages

The PIC processors use buffers to store incoming data from the XBee modules. If the buffers fill up too quickly, data packets can be lost. A way to ensure this doesn't happen could be beneficial to ensure the status of tags is not inaccurate.