Global Scientific JOURNALS
GSJ: Volume 10, Issue 1, January 2022, Online: ISSN 2320-9186
www.globalscientificjournal.com

WIRELESS PORT DESIGN FOR SENSOR AND ACTUATOR NETWORKS

Sanskar Biswal
sanskarbiswal@gmail.com
School of Electronics Engineering (SENSE), Vellore Institute of Technology (VIT), Vellore

Abstract:
In this paper, we illustrate the design process and schematics of a new design of wireless-node port, having the ability to interface with sensors and actuators alike. It works on the principle of a master-slave network, with each node acting as a slave to the master connected to a cloud-based monitoring system. We focus on the design paradigms to develop a fully capable wireless-node. The aim is to have the designed node be as power-efficient as possible and have maximum configurability without the need of any form of hardware changes. Thus, the node needs to be able to be configured via code, while consuming minimum resources. An Embedded System was developed as a result, which uses an ATtiny85 microcontroller and nRF24L01 module for communication. A simple table of variables stored in the EEPROM of the microcontroller, can fully be modified to suit the type of operation needed by the device. The values of this table may be modified via RF communication from the master-node, thus making the same hardware design feasible and adaptable for both sensor networks, actuator network and even hybrid networks.

Index Terms: ATtiny85, RF Communication, nRF24L01, Wireless Sensor Networks, Wireless Actuator Network, Hybrid Wireless Networks, Self-Adjusting Topologies, EEPROM Programming Tables, SPI Protocol

1. INTRODUCTION

Sensor networks often have a dedicated hardware that are custom built for specific sensors. Thus, any major operational change often involves the overwriting of the device/node firmware and even perhaps substitution of the hardware with another. This process not only adds to the delay in deployment of changes, but also considerably increases the cost of changes. This problem is compounded by several magnitudes with increasing scale of deployment. It is with this problem in mind that this project was conceptualized.

The advent of IoT has made it very easy to perform complex scale processing and decision-making on a server. This is a novel solution for constraints due to local factors, however this also inevitably introduces latency into the system. In many real-time systems, reaction to changing conditions may be a critical factor, not wholly dependent on any decision from a cloud-based master code. In such scenario, rapid inputs from the environment must result in immediate reaction from the system, with minimum latency. Thus, organizing a topology where the sensor sends data directly to the actuator/network of actuators is a potent solution. There is however the chance that the configuration of the sensor or the network needs to be modified. Such scenarios may again lead to excessive delays if there is a need to develop custom hardware or firmware for each use-case.

The proposed solution will aim to mitigate the above-mentioned challenges, by designing a compact unit which has the provision to connect to a variety of sensor and actuator configurations, but also provide a method of wirelessly communication amongst the relevant nodes. A major advantage of wireless networks is that they do not have the need of regular maintenance. To ensure that this holds true for the proposed solution, the design will also need to be as power-efficient as possible.

Thus, the objectives of the project may be broadly listed as the following:

- A compact system, which allows for digital and analog I/O
- Wireless communication module, with configurable addresses to facilitate dynamic topologies

- Power efficient design of the system, to allow the module to function on a battery for the maximum possible duration

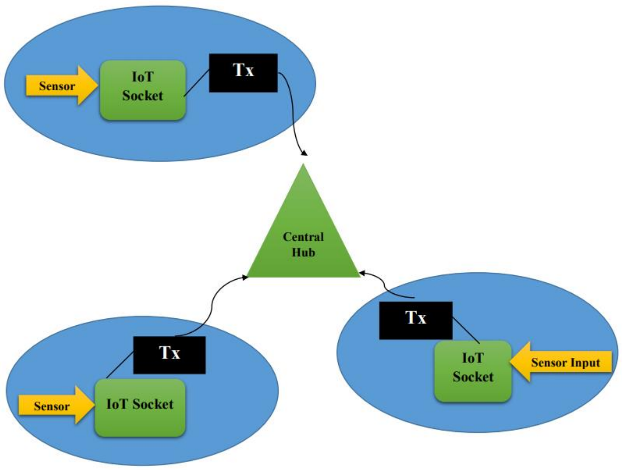

Fig. 1. Networked-Node Topology

1.1. Microcontroller and RF Module

The microcontroller for the designated node opted in this project is the ATtiny85. ATtiny85 is a very compact and power-efficient chip which belongs to the Atmel family of microcontrollers. It has very good documentation [3] and a wide base of supporters in the developer community as well.

The primary reason for the selection of the ATtiny85 is due to its power-efficiency. In the datasheet [3], it is stated that the ATtiny85 consumes 300µA in active mode and 0.1µA in sleep-mode. This is an ideal feature given the requirements.

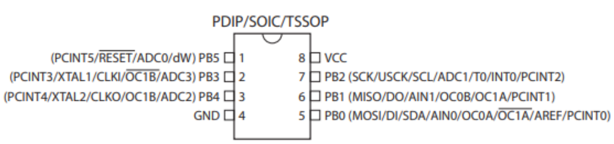

Fig. 2. ATtiny85 Pin Configuration

In the figure above, it may be seen that ATtiny85 has a total of 8-pins, out of which, two are power pins (VCC/GND). This leaves us with 6 pins for I/O. Amongst these, pin1 functions as the reset pin and is hence reserved for that purpose. Pin5,6,7 respectively are the MOSI, MISO and SCK pins. These pins are needed to ensure serial communication with the RF module and hence cannot be used for I/O. Thus, there are only 2 pins remaining, pin3 and pin2 for I/O. A solution to the restricting number of pins maybe the use of a microcontroller with a larger number of I/O pins. However, the aim of the project is to design the most energy and space efficient module possible. Thus, the ATtiny85 module is still the processor of choice. The method of mitigating the small number of I/O pins is discussed in the **Design** section of the paper.

The nRF24L01 module is the chosen for the purpose of RF communication. RF communication is used for a variety of reasons, including but not limited to its range of communication extending up to 100m and low power consumption. nRF24L01 module in specific utilizes the 2.4GHz band for communication and used 1MHz bandwidth for each channel. This setup allows up to 125 simultaneous channels to be utilized in a limited space. The 2.4GHz frequency is very permeable through physical boundaries, and thus do not need line-of-sight setup for operational efficiency.

nRF24L01 module consumes an average power of 13mA in its active mode of operation, which is the primary operating mode. Thus, the remainder of power consumption will be either from the actuator or the sensor.

It can thus be assumed that the net power consumption of the fully operational system, including the sensor units will be around 20mA. Actuators will mostly require external power source for operations. The reason for this may be attributed to the working voltages for the microcontroller and the RF module, both of which are 3V. Although they can work at higher voltages, the RF module has a tendency of increased failure. Thus, to prolong the lifetime of the module, 3V DC will be the preferred method to power the module. In all further references to the module will be classified broadly as Node. This is because the Node is in principle, a component of a tree like topology.

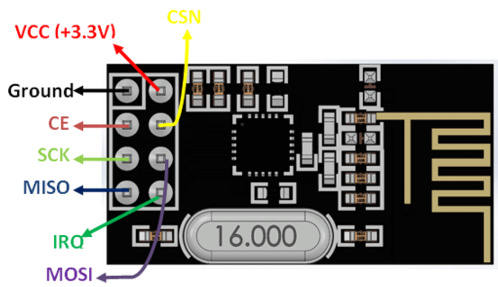

Fig. 3. nRF24L01 pin configuration

The datasheet of the nRF24L01 [4] module specifies that it uses SPI communication protocol for the communication of data to and from the module. The MOSI of RF module connects to the MISO of the microcontroller and vice-versa. The SCK pin connects to the SCK pin of the microcontroller. CE and CSN are Chip Enable and Chip-Select-Not respectively. They can both be set too high to always have the module turned on. The IRQ is the interrupt pin which does not need any form of connection in the present design.

1.2. Wireless Topologies

The present paper aims to design a module that can operate in the following topologies:

- Sensor Networks
- Actuator Networks
- Hybrid Networks

Sensor networks, as the name suggests comprises fully of sensors. The type of sensors may vary, but the nodes only send collected data, with timestamps. The only limitation is that the number of Input ports can be a maximum of two, accounting for the number of I/O ports on the microcontroller. This however, from the study in [5] may not necessarily be limitations as there are very few sensors that have a higher number of input ports. Cameras and combined sensors are the only sensors that may exceed the port capacity. Most other sensors will work well the current design.

Actuator Networks comprise mostly of output modules like motors and speakers. Mechanical and Audio are the two outputs that can be handled by the ATtiny85 without any issues. Display output is not possible due to the port limitation. However, most tasks at home/industrial environments can be easily handled with the two I/O pins. The two pins may also be used to output control signals to the environment, the frequency and the magnitude of which may be set by either the master-node or may be a preset value.

Hybrid networks comprise of both sensor and actuators at the node. These nodes need not be connected to the parent, but instead to a sensor in its operational requirement, thus forming a closed-loop feedback system as proposed in [12].

A hybrid network is more closely related to a real-world operating scenario and is better equipped to perform at more optimal levels when compared to homogeneous element networks. This observation is very commonly seen where the action of an actuator is closely dependent on the inputs from the direct environment. Examples may include self-balancing bots and automated door opening system. All are essentially a hybrid system. A network of moisture sensors may act as the input for opening of floodgates in a water reservoir, which may again be classified as hybrid network.

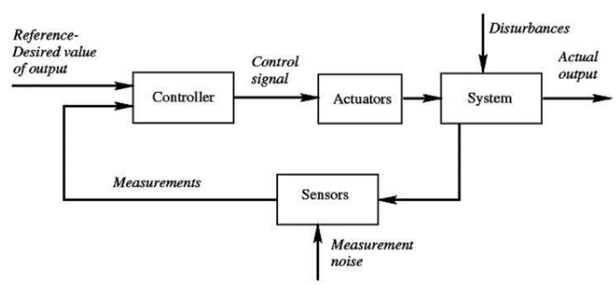

Fig. 4. Sensor-Actuator Feedback System Model

2. Design Procedure

The design process depends mainly on the objectives established at the onset of the paper. The designed Node need not have high processing power as its main purpose is to provide an I/O port for sensors or actuators and a communication module to establish a network of connected Nodes.

It has been already established earlier that the ATtiny85 will have only 2 pins available for I/O after accounting for SPI connectivity. The most optimal method for the use of two pins is have them for I/O when needed.

Thus, any sensor/actuator connected to the Node will have connections directly to I/O pins from their O/I pin. Power to the connected module will be provided from the power source of the Node. This power module is limited to 3V, thus motors will normally require external power supply, to avoid a faster drainage of the battery capacity of the Node. In case of motors, the Node may be used as a switch, to control the rate of operation, thus increasing the working life of the node by a considerable margin.

2.1. Battery and Power Systems

The developed power system needs to be capable of providing power to the Node at least one year to be considered a long-term operational module.

We know from Section 1.1 that the expected power consumption of the system will be at the maximum threshold of 20mA/s. Thus, the power consumption of the node is 20mW.

Battery Life = Battery Capacity in mAh / Load Current in mA * 0.70

i.e.: Capacity = Battery-Life*Load*0.7

Battery life in the above equation is measured in hours. Thus, in one year there are 8760 hours.

$$8760\text{hrs} * 20\text{mA} * 0.7 = 122640 \text{ mAh}$$

The result presented by the equation above indicates the need to use a battery pack with almost 1.25k-mAh capacity. Such a battery system is not only prohibitively large and costly, but mainly impractical. The solution to this was to either power the Node directly from a continuous power supply or to reduce the net consumption of the device.

It is not possible to reduce the power consumed by the device directly. However, it is possible to reduce the uptime/active-time of the node. In the scenario where the node is activated only once every 10s, then the required capacity drops to 12000mAh and though this is not small, it is still feasible. However, if the active time of the Node is reduced further to once every minute, the required capacity drops to 2000mAh which is a very realistic and feasible option for utilization.

At this point, the question arises regarding the feasibility of using a system that can send data only once every minute. The case is analysed from two perspectives, drawing inspiration from results and observations drawn from [9].

In home/domestic scenarios, most of the sensors do not interact in an environment that changes rapidly, thus making almost no impact on the reduced time of access. For system, that require full up-time, they will need to be connected to an external source. Nodes that connect to actuators can be powered from the power source of the actuators. To account for the variety of input voltages, a safety circuit needs to be built to reduce chances of the Node failing due to power fluctuations.

In industrial scenarios where there is the need for more frequent control, the networked mode of operating may be used. Since industries often operate in a large scale, it reasonable to assume that there will be several sensors/actuators connected to the master. These may operate at a cyclic period in such a fashion that there is always of set of Nodes active at each second. The scale of operation will offset the limited uptime of the Node.

This solution may be unconventional but has the potential to work in several use-cases, some of which will be tested in simulations.

Thus, we assume that the required capacity for one-year operation is 2000mAh. Via the comparison of various cell costs and capacities, AA-cells are the best option. They have a potential difference of 1.5V and a rated capacity of 1500mAh. Batteries in Series connection will add-up in capacity, where as in Parallel, they add up in voltage. Thus, a combination of two AA-cell in series in parallel, will yield a voltage of 3V at 3000mAh capacity which sufficiently satisfy the required objective. Thus, each module may be powered either from an external power supply or from 4 AA-cell combination battery pack.

2.2. I/O and RF Module Connections

The I/O is supported from two pins on the ATtiny85. A common design in most system is to have sensors ports have either a 4 or 3 pin setup. Two of these pins are the VCC/GND, and they are usually at the extreme ends of the sensor module. The control/data pins are at the centred on the sensor module. This has been utilized to make the connector port accessible to a wide variety of sensors. The GND of the sensor connects directly to the GND of the ATtiny85. It is followed by the 2-pins for I/O and then the Reset-Pin.

If the sensor is a 3-pin module, Pin-2 of the ATtin85 is used as the VCC but having that pin set to HIGH (3V). Thus, Pin-3 will be connected to the I/O of the sensor.

If the sensor is a 4-pin module, Pin-3 and Pin-4 connect to the I/O, whereas the VCC of the sensor connects to the VCC from the power-source.

We thus will need two separate ports, one each for 3- and 4-pin sensor modules.

Output devices are easier to interface as most have only one pin for control/output.

The nRF24L01 as mentioned in [11] has a much simpler connection to the ATtiny85. The MOSI -> MISO, SCK -> SCK connection scheme is very straight-forward. The main design needs to be worked out in the activation of the module as it is the most power consuming unit on the node. The CE pin of the nRF24L01 module can be set to LOW, to turn-off the module. However as mentioned in [6], the power consumption of the RF module is 900μA in standby mode and only during transmission or reception does it consume 13mA of power. It thus is of almost no consequence if the RF module is always on as it will spend most of its time in standby mode of operation. The following table is taken from [6] indicating the power consumed by the nRF24L01 module in various modes of operation.

Parameter (condition)	Notes	Min.	Typ.	Max.	Units
Idle modes					
Supply current in power down			900		nA
Supply current in standby-I mode	a		26		μA
Supply current in standby-II mode			320		μA
Average current during 1.5ms crystal oscillator startup			400		μA
Transmit					
Supply current @ 0dBm output power	b		11.3		mA
Supply current @ -6dBm output power	b		9.0		mA
Supply current @ -12dBm output power	b		7.5		mA
Supply current @ -18dBm output power	b		7.0		mA
Average Supply current @ -6dBm output power, ShockBurst™	c		0.12		mA
Average current during TX settling	d		8.0		mA
Receive					
Supply current 2Mbps			13.5		mA
Supply current 1Mbps			13.1		mA
Supply current 250kbps			12.6		mA
Average current during RX settling	e		8.9		mA

Fig. 5. Power Consumption in nRF24L01

Thus, the CE and CSN pins are connected to the power source of the Node. The IRQ pin is left unconnected to any unit on the module. A simple voltage control circuit is connected to the CE/CSN pins of the RF module to prevent any excess voltage from affecting the operating of the module. The operating voltage range is up to 3.6V and although the nRF24L01 module supports up to 5V, this circuit will help prevent any unintended module failure due to over-voltage.

2.3. Schematic and PCB Development

The schematic for the design and development made use of the Autodesk Eagle software. These schematics may be accessed at the link provided in [7].

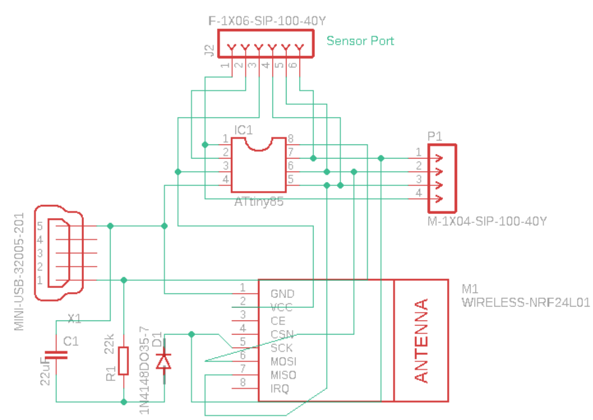

Fig. 6. PCB Schematic

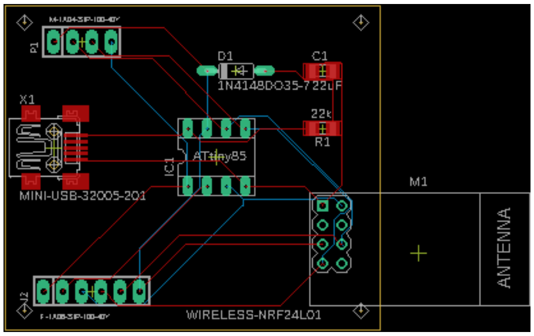

Fig. 7. PCB Layout

3. Firmware Design

The firmware/code on the module is the most essential and critical component. Since the module has a pre-set hardware, the variety of sensors and actuators will rely on efficient manipulation offered by the firmware for interfacing.

There are 4 major operations that need to be handled in any update:

1. Address Update
2. Pin Configuration Update
3. Transmit Condition
4. Transmit Time Update

Address update is used to store the address of the node. Each Node will have its address on itself, and the parent-hub will store all addresses at a local table. Changing the address of the Node provides the user with the flexibility to modify the topology of the network.

Pin configuration, for this specific use-case needs to set if the pin performs an input or output operation and if the I/O is digital or analogue in nature. There are 2 I/O pins free in the module and thus, 4 variables are used to freely program this configuration set. A set of simple flags (0/1) are used to indicate the configuration.

The transmit condition is used to program the data transmission condition. It can either be periodic or event based, in which case it is to be mentioned which pin I/O is the event pin.

Transmit Time Update, is the transmission interval for periodic transmit condition.

3.1. Data Structure

The module uses C structures to enable communication between the parent and the various nodes. The main structure which is transmitted is called mainConfig which comprises of opCode, which is the integer number of the operation to be carried out.

Each operation has its own structure which is a member of the mainConfig structure. The Nodes, will receive the entire structure and run a switch case in the received opCode. Functions written on the Node then process the configurations and then run the necessary processes to update the EEPROM table. Once this is done, the system is reset to run operations with the updated configuration.

```
struct txData {
  unsigned long p3Data;
  unsigned long p2Data;
};
// Structure of I/O pin configuration
struct ioConfig {
  byte p3Mode [1];
  byte p2Mode [1];
  int p3Dir;
  int p2Dir;
};
// Structure of Rx Data
/*
 * opCode = 1 => Address Update
 * opCode = 2 => Pin Update
 * opCode = 3 => Time Config
 * opCode = 4 => Time Period Update
 */
struct rxData {
  int opCode;
  ioConfig pinConfig;
  uint16_t addr;
  unsigned long t_period;
};
```

The sensor data is wrapped in txData and delivered to the parent-hub, which then unwraps the structure to finally access the data.

TABLE 1
EEPROM Lookup Table

Address	Config
0x398	Init (ref. for first run)
0x400	Address
0x420	t_period {for T_Config = 0}
0x424	Pin3-Data-type (Digital/Analog)
0x425	Pin2-Data-type (Digital/Analog)
0x426	Pin3-dir (Input/Output)
0x428	Pin2-dir (Input/Output)
0x430	T_Config (0/1/2/3) {1=> both}

4. Testing and Results

Functional Loop Structure

1. Initialize PIN from EEPROM
2. Initialize RF Module
3. Get sensor Data
    a. If data is analog -> ADC => Digital Mapping
    b. If data-dir is Input, read data
    c. Else Output Value
4. Stop Listening for Signals
5. Send data using data structure via RF Module
6. Start Listening for Signal
7. If data is available from RF radio
    a. Read data into structure
    b. Switch on opcode
    c. Run opcode command sequence function
    d. Call setup to restart module with new command format

The module was tested with a PIR sensor, the data pin at pin3 of the ATtiny85. The collected data was observed on the Serial Monitor at the parent-hub. The data is at pin3 and pin2 sends an increasing counter as a default setting.

Serial Output at Parent Hub:

Received: 1
Pin 3 = 1
Pin 2 = 100

Received: 1
Pin 3 = 1
Pin 2 = 101

Received: 0
Pin 3 = 0
Pin 2 = 102

Updating the config needs to be done via the Serial Monitor. However, manual input to the Serial Monitor does not confirm to the format of the data structure. Thus, as a potential improvement, an UI can be developed to select the pin configurations and send the data in the right format to the Node.

The output mode was tested using a regular DC motor and all functions were observed to be consistent with the predicted outcomes.

5. Conclusion

The results obtained from the test conducted in Section 4 indicate the feasibility, validity and utility of the module designed.

We can conclude the design and the codes for the Node are operational and functional. There remains scope to further improve the project via the addition of UI and a more robust security feature for the Nodes connected in the network. Discussion on those topics are beyond the scope of this paper and are left for future development and design.

6. Acknowledgements

This work was supported by the faculty and staff at Vellore Institute of Technology, Vellore. We would like to extend our gratitude to the management of VIT, Vellore and Dean of School of Electronics Engineering for their constant support during the process of design and development of this project.
This project was conducted under the guidance of Prof. Yepuganti Karuna (karun@vit.ac.in).

7. References

[1] Zeeshan Hamid Mir, Young-Bae Ko, "Collaborative Topology Control for Many-to-One Communications in Wireless Sensor Networks"
DOI: https://doi.org/10.1109/ACCESS.2017.2722379
[2] Hium Kim, Abbas Ahmad, Hamza Buqa, Jaeyoung Hwang, "IoT-Taas: Towards a Prospective IoT Testing Framework"
DOI: https://doi.org/10.1109/ACCESS.2018.2802489
[3] ATtiny85 Datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf , 1
[4] nRF24L01 Datasheet: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf
[5] https://www.electronicshub.org/different-types-sensors/, Types of common DIY Sensors
[6] nRF24L01 Datasheet, Section5.1, Power consumption, page-14
[7] IoT Connect Files and Schematics: https://github.com/sanskarbiswal/IoT_Connect
[8] James Pope, Robert Simon, "Efficient one-to-many broadcasting for resource-constrained wireless networks"
DOI: https://doi.org/10.1109/LCN.2015.7366366
[9] Halah Mohammed Al-Kadhim, Hamed S Al-Raweshidy, "Energy Efficient and Reliable Transport of Data in Cloud-IoT"
DOI: https://doi.org/10.1109/ACCESS.2019.2917387
[10] nRF24L01: In-Depth: How nRF24L01 Wireless Module Works & Interface with Arduino
[11] RF module and ATtinty85 connection: nRF24L01+ with ATtiny85 3 Pins - Hackster.io
[12] Yong Feng, Hai Liu, Jie Yang, Xiaodong Fu, "A localized inter-actuator network topology repair scheme for wireless sensor and actuator networks"
INSPEC Accession Number: 18510119