IOT Challenge #1 report

Francesco Spangaro - Luca Tosetti

20 March 2025

POLITECNICO
MILANO 1863

Academic Year 2024 - 2025

Contents

1 Introduction

We were tasked with developing and emulating a parking sensor. This IOT device uses the HC-SR04 ultrasonic distance sensor in order to evaluate the distance between itself and the car parked on top of it, and the ESP-NOW for WiFi communication. The device has to periodically check the occupation of the parking spot it is assigned to: once every duty-cycle. After the execution is finished, the board is set to go into deep sleep mode with a wake-up timer of X seconds. When the device is not measuring or sending data, it must be put in deep sleep. We must perform an estimation of the power and energy consumption of a full duty-cycle, estimating the amount of time the device can run on a single charge of its battery. The battery has a total energy of Y. Both Y and X are evaluated using the group leader's personal code (10734844).
$X = 44 \ \% \ 50 + 5 = 49$
$Y = 4844 \ \% \ 5000 + 15000 = 19844$
With % being the modulo operator.

2 Schematics

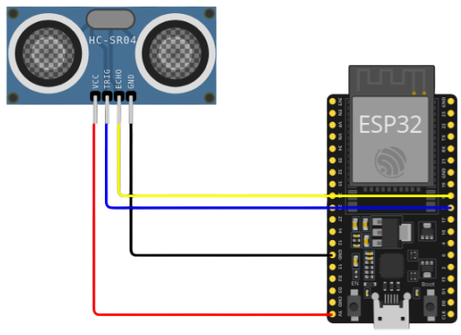

Figure 1: System's circuitry

3 Software definition

3.1 Setup

In the software's setup section, we prepare the board for the computation that will happen in the software's loop section. We set the WiFi module to off and prepare the ESP32 board for a new computing cycle. Here we also evaluate the time required to correctly configure the module and start preparing for measuring the other needed times.

3.2 Loop

We have divided the loop section into multiple steps, in order to:

1. Obtain the distance between the device and the car on top of it.
2. Send to the sink node a string, informing it if the parking spot assigned to the device is either occupied or free.
3. Go into deep-sleep mode.

3.2.1 Distance measurement

In the performDistanceRead() function firstly we reset the TRIG_PIN to low, to make sure that the measurement will be done precisely. Then we set the TRIG_PIN to high and wait for 10 microseconds, in order to let the HC-SR04 module measure the distance between the IOT device and any car that could be on top of it. We then reset the TRIG_PIN to low. The HC-SR04 sensor evaluates the time it takes for ultrasounds to bounce off the first item in front of it; so to properly evaluate the distance between the sensor and any item, we divide the ultrasound's trip time by 2 and multiply it by the speed of sound. This gives us the distance measurement in cm.

$$distance = time * \frac{SOUND\_SPEED}{2}$$

If the distance is less than 50 cm, we can agree that the parking spot is occupied, otherwise it is free.

3.2.2 Result's communication

To communicate the result obtained, we connect via WiFi to a "Sink node", a singular node in the network tasked with receiving data from all sensing nodes. To do so, in the sendMessage(String message) function we enable the WiFi module. Then, using the ESP-NOW protocol, we send to the sink node the message: "OCCUPIED", if the distance between the IOT device and the first object on top of it is less than 50 cm, "FREE" in the opposite case. Once we send the message, we disable the WiFi module by setting it to its "off" state.

3.2.3 Duty-cycle time evaluation

While the board is executing the code, we evaluate the time it takes for every single macro-function in the code to complete. All the computed times are:

- Setup time: time taken by the device to setup, before the computation, average: 0.88 ms;
- Sensing time: time taken by the device to evaluate the distance between itself and any object on top of it, average: 24.0 ms;
- Transmitting time: time taken by the device to transmit the result ("FREE"/"OCCUPIED") to the sink node, average: 188.63 ms;
- Execution time: sum of the above three times;
- Deep-sleep time: time the device spends in deep-sleep mode, always equal to X seconds, average: 49.0 s;
- WiFi on time: time the device spends with the WiFi module enabled (it coincides with the transmission time, since the WiFi module is enabled only during the transmission), average: 188.63 ms;
- WiFi off time: time the device spends with the WiFi module disabled (coincides with the sum of the sensing time and setup time since the WiFi module is disabled at each cycle start and remains disabled until the transmission phase), average: 24.88 ms.

4 Energy consumption estimation

To evaluate each phase's power consumption we use the formula:

$$E_{c,\ phase} = t_{avg} \ * \ P_{avg}$$

4.1 Deep-sleep phase

From the data found in the deep-sleep power consumption file, we evaluated four average power consumptions, one for each phase:

- Deep-sleep phase: average 59.66 mW;
- Wake-up phase: average 367.14 mW;
- WiFi enabled phase: average 775.49 mW;
- WiFi disabled phase: average 310.97 mW;

For each phase, we found the total power consumed by the device:

- Deep-sleep phase: $E_{c,ds} = 49s \ * \ 59.66mW = 2.9234J$;
- Wake-up phase: $E_{c,wu} = 0.88ms \ * \ 367.14 \ mW = 0.0003J$;
- WiFi enabled phase: $E_{c,WiFi\ en} = 188.63ms \ * \ 775.49 \ mW = 0.1463J$;
- WiFi disabled phase: $E_{c,WiFi\ dis} = 24.88ms \ * \ 310.97 \ mW = 0.0077$;

POWER CONSUMPTIONS		
Measure	mW	W
DEEP SLEEP		
Deep_sleep_avg	59,66	0,060
Deep_sleep_wakeup_avg	367,14	0,367
Deep_sleep_WiFi_on_avg	775,49	0,775
Deep_sleep_WiFi_off_avg	310,97	0,311

Figure 2: Deep-sleep power consumptions

AVERAGE TIME DURATIONS			ENERGY CONSUMPTION
Phase	ms	s	J
Wakeup	0,88	0,0009	0,0003
Deep_sleep	49000,00	49,0000	2,9234
WiFi_on	188,63	0,1886	0,1463
WiFi_off	24,88	0,0249	0,0077

Figure 3: Deep-sleep phase times

4.2 Sensor reading phase

From the data found in the sensor reading power consumption file, we computed two average power consumptions:

- Sensor reading phase: average 466.74 mW;
- Sensor idling phase: average 331.59 mW

We then found the energy consumed in each phase by the sensor:

- Sensor reading phase: $E_{c,sr} = 24ms \ * \ 466.74mW = 0.0112J$;
- Sensor idle phase: $E_{c,si} = 189.50ms \ * \ 367.14 \ mW = 0.0628J$;

POWER CONSUMPTIONS		
Measure	mW	W
SENSOR		
Sensor_read_avg	466,74	0,467
Sensor_idle_avg	331,59	0,332

Figure 4: Sensor power consumptions

AVERAGE TIME DURATIONS			ENERGY CONSUMPTION
Phase	ms	s	J
Sensor_read	24,00	0,0240	0,0112
Sensor_idle	189,50	0,1895	0,0628

Figure 5: Sensor phase times

4.3 Transmission phase

From the data found in the transmission power consumption file, we computed three average power consumptions:

- Transmission idle phase: average 704.22 mW;
- Transmission phase on 2dBm: average 797.29 mW;
- Transmission phase on 19.5dBm: average 1221.76 mW;

We found the energy consumed by the device only when the transmission is active on 2dBm, since we assume that the sink node is close to the sensors enough to receive the data without needing to activate the 19.5dBm transmission:

- Transmission phase on 2dBm: $E_{c,2dBm} = 188.63ms \ * \ 797.29 \ mW = 0.1504J$;

POWER CONSUMPTIONS		
Measure	mW	W
TRANSMISSION		
Tx_idle_avg	704,22	0,704
Tx_2_dBm_avg	797,29	0,797
Tx_19.5_dBm_avg	1221,76	1,222

Figure 6: Transmission power consumptions

AVERAGE TIME DURATIONS			ENERGY CONSUMPTION
Phase	ms	s	J
Transmission	188,63	0,1886	0,1504

Figure 7: Transmission [2dBm] time

4.4 Final evaluations

After computing the energy consumed by each phase, we evaluated the total energy consumption for a single duty-cycle. With that, by dividing the battery's available energy with the energy consumed by one duty-cycle, we are able to estimate the average amount of complete duty-cycles the device can perform before the battery dies.

$$\#cycles = \frac{E_{battery}}{E_{dc}}$$

The total number of cycles the device can perform on one battery charge is:

$$\#cycles = \frac{19844J}{3.30J} \simeq 6009$$

Which amounts to approximately 3 days and 6 hours of work.

AVERAGE TIME DURATIONS			ENERGY CONSUMPTION
Phase	ms	s	J
Wakeup	0,88	0,0009	0,0003
Sensor_read	24,00	0,0240	0,0112
Transmission	188,63	0,1886	0,1504
Deep_sleep	49000,00	49,0000	2,9234
WiFi_on	188,63	0,1886	0,1463
WiFi_off	24,88	0,0249	0,0077
Sensor_idle	189,50	0,1895	0,0628
Total	49213,50	49,2135	3,30

Figure 8: Total phase times evaluation

Person code	Battery capacity	Duty cycle duration
10734844	19844	49

Estimated num of cycles	Battery duration
6009	295724 s
	82 h
	3,4 days

Figure 9: Duty-cycle count evaluation

5 Remarks

The system, after computing its lifetime and power consumption, turns out not to be practically usable in a long-term application scenario, since each sensor would need its battery replaced every three days. This system is well designed for light uses, for example, for setting up a parking spot counter for events that last a weekend or less. To further increase the system's lifetime, three possible improvements can be considered.

6 Improvements

6.1 Increase the deep-sleep duration

For a device that checks whether a car has been parked on it or not, a period of $49s$ between two measurements is too small. We think that using two different deep-sleep times could be ideal. We would sense the occupation of the parking spot once every $49s$ if and only if it is found to be free. Otherwise, if the parking spot is occupied, we would change the deep-sleep time to an educated guess on the average amount of time a person stays at the event. E.G.: $5min$ for a supermarket parking spot, $1hr$ for a wedding venue.

6.2 Increase the battery capacity

The device's battery capacity is $19844J$, which is only a small percentage increase over a simple AA battery. By changing to a lithium battery, which has a capacity that averages $172kJ$, we could increase the lifetime of the system ten fold, going from a three-day lifetime to almost a month of continuous work.

6.3 Communicate to the sink node only on status changed

The way the system is currently implemented, every $49s$ a measurement is taken and the resulting status is sent from each sensor to the sink node. The transmission period is the most time- and energy-consuming part of the device. If we instead persist the last sensed occupation status between duty-cycles, we could avoid the message transmission between the sensors and sink node if the newly measured status is equal to the last sensed one. This would greatly decrease the overall time- and energy-consumption of the system, lengthening its lifetime.

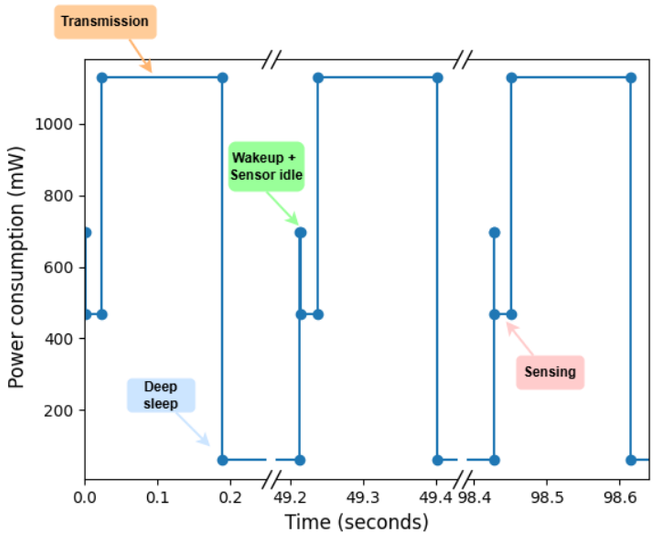

Figure 10: System energy consumption per duty-cycle

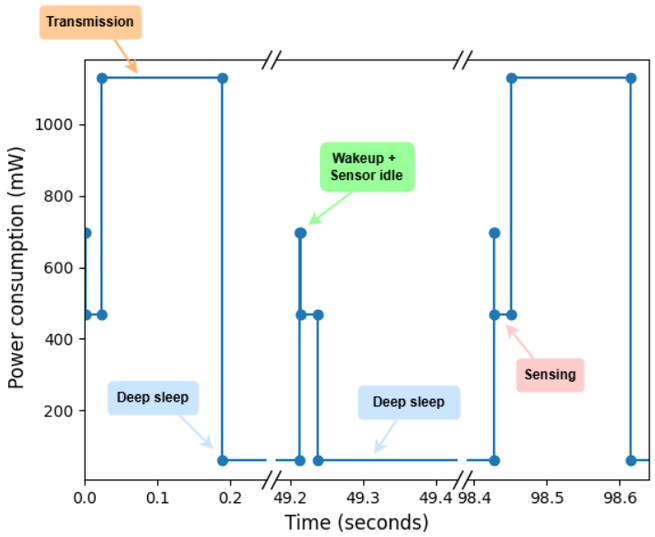

Figure 11: System energy consumption per duty-cycle after improvement 6.3