IoT Project Evaluation Criteria

ESSENTIAL REQUIREMENTS

- Wireless interface between the buried sensor node and the device attached to the hand held device.
- The translating device must use the power from the hand held device (no external battery).
- User friendly data visualiser on the hand held device.
- Low energy consumption of the translating device.
- Documentation and presentation of the work.

EVALUATION POINTS

- Implementation with least number of components
- Low power consumption
- Successful data reception and visualiazation
- User friendly graphical user interface
- Cost effective solution

All the projects will be evaluated on the above mentioned points. The group that presents the most optimised and user friendly solution will be given the best IoT project award.

EVALUATION SCENARIO

Every group will be asked to receive the data transmitted from a MoleNet sensor node. Each group has to store and visualize the received data. The underground MoleNet sensor node will sample and transmit packets at different frequencies i.e. burst of packets together, one packet every minute, one packet every 5 minutes, etc.

The underground MoleNet sensor node will send packets of length 23 bytes. Structure of the packet transmitted is shown below.

[**PACKET_SENT** 1=>254] { **HEADER** [**sinkID** 254] [**sourceID** 1] HEADER**}
{ **DATA_PLUS_INFO** [**Length** 23] [**sourceID** 1] [**RTC** 548965593 25/5/17 18:26:33] [**dielectric** 0] [**temperature** 0] [**pktSent** 75] [**reteries** 0] [**pktLOST** 46] [**RTT** 7] [**RSSI** 0] [**nextAddr** 552] **DATA_PLUS_INFO** }

Different parameters can be extracted from the packet received by the gateway node i.e. parameters **dielectric** and **temperature** indicate the values of the sensors connected to the MoleNet sensor node. The packet transmitted can be received and relevant information can be extracted. The listing below shows the position of parameters in the packet received.

- pos_PACKET_TYPE,
- pos_PACKET_LENGTH,

- pos_SOURCE_ID,
- pos_EPOCH_0,
- pos_EPOCH_1,
- pos_EPOCH_2,
- pos_EPOCH_3,
- pos_DIELECTRIC_0,
- pos_DIELECTRIC_1,
- pos_TEMP_0,
- pos_TEMP_1,
- pos_PACKETS_SENT_0,
- pos_PACKETS_SENT_1,
- pos_SENDING_RETRIES,
- pos_PACKETS_LOST_0,
- pos_PACKETS_LOST_1,
- pos_RTT_0,
- pos_RTT_1,
- pos_RSSI_0,
- pos_RSSI_1,
- pos_EEPROM_CURRENT_PAGE_0,
- pos_EEPROM_CURRENT_PAGE_1,
- pos_EEPROM_CURRENT_PAGE_2,

All the parameters should be extracted and stored on the gateway. The two sensors data at least should be visualised on the application running with the gateway. The other data can be used for more user-friendly representation of the sensors' data.

Every group can test their implementation against the required criteria before the final competition on June 8th during class.
During the competition, you will have at most 3 re-tries to start your gateway device (after erasing the data on it) and get it running. However, the last re-try will start at 10:30.

Power consumption evaluation

In order to evaluate the power consumption of the gateway, we will use a two-fold approach. Before starting the competition, you need to provide us with:

1. A list of all hardware components on your gateway and their exact energy consumption in all relevant modes.

2. An evaluation of the duty cycle of each component.

3. Copy of the preliminary code of your project. It will be only used to validate the duty cycles, not for evaluating the quality of the code!

A template list is provided at the end of this document and can be used directly. One of the last two columns must be filled in, depending on the implementation you have. **We encourage you to prepare the list for June 8th, so you can get feedback on it.**

User friendliness evaluation

Every person in the room will be given one vote to give to the project, whose user friendliness she/he considers best. You can use your vote as you see fit, but you cannot vote for your own project. Teachers have also one vote per person.

Table 1: Power consumption details

Component (exact specification)	Mode	Power consumption [mA]	Percent time in mode [%]	Call every X [secs]
Microcontroller	on			
	sleep			
	deep sleep			
Radio Transceiver	send			
	receive			
	sleep			
External memory	write			
	read			
	sleep			
RTC				