2º Hackathon SEMISH

DESAFIO Venha construir soluções de IoT utilizando a plataforma KNoT

XXXVIII Congresso da Sociedade Brasileira de Computação
#ComputaçãoeSustentabilidade
22 a 26 de julho | Centro de Convenções | Natal-RN

Realização:

Apoio:

KNoTThing API

The constructor creates a KNoT Thing object, that is responsible for handling the read/write operations over the sensors and actuators. It also runs a state machine for managing the Thing states during its life cycle (registering/authenticating/handling messages/etc).

```
KNoTThing();
```

This method initializes logging, radio, storage and all other modules. It also registers the Thing name to be used (announced) by the protocol. This function must be called on setup().

```
int init(const char *thing_name);
```

This function is responsible for running the Thing state machine. It will connect to the gateway, register/authenticate the Thing, send the Thing description and wait for messages. It also sends messages with sensor data, according to each sensor configuration (see the registerDefaultConfig function). This function must be called on loop().

```
void run();
```

Functions that are responsible for registering all sensors and actuators for this Thing. By registering, the KNoT Thing object is able to read the sensors and write on actuators in response for the messages from the gateway.

```
int registerIntData(const char *name, uint8_t sensor_id, uint16_t type_id,
uint8_t unit, intDataFunction read, intDataFunction write);

int registerFloatData(const char *name, uint8_t sensor_id, uint16_t type_id,
uint8_t unit, floatDataFunction read, floatDataFunction write);

int registerBoolData(const char *name, uint8_t sensor_id, uint16_t type_id,
uint8_t unit, boolDataFunction read, boolDataFunction write);

int registerRawData(const char *name, uint8_t *raw_buffer, uint8_t
raw_buffer_len, uint8_t sensor_id, uint16_t type_id, uint8_t unit,
rawDataFunction read, rawDataFunction write);
```

The definition of the read and write functions are shown below:

```
typedef int (*intDataFunction)(int32_t *val, int32_t *multiplier);
typedef int (*floatDataFunction)      (int32_t *val_int, uint32_t *val_dec,
int32_t *multiplier);
typedef int (*boolDataFunction)       (uint8_t *val);
typedef int (*rawDataFunction)        (uint8_t *val, uint8_t *len);
```

This function is responsible for defining the default behaviour (the conditions that the Thing will send messages) for each sensor/actuator. The conditions are defined by the event_flags parameter and could depend on: time, value change, min or max thresholds.

```
int registerDefaultConfig(uint8_t sensor_id, uint8_t event_flags, uint16_t
time_sec, int32_t upper_int, uint32_t upper_dec, int32_t lower_int, uint32_t
lower_dec);
```

The **event_flag** parameter define when (or why) you want it to send information to the cloud:

- By periods of time (KNOT_EVT_FLAG_TIME).

- When the value is greater than a threshold (KNOT_EVT_FLAG_LOWER_THRESHOLD).

- When the value is lower than a threshold (KNOT_EVT_FLAG_UPPER_THRESHOLD).

- When the value changes (KNOT_EVT_FLAG_CHANGE).

- Combinations of these (sum).

Flag Alias	**Flag Value**	**Description**
KNOT_EVT_FLAG_TIME	1	Send data every period of time, in seconds. Needs a value greater than 0 to be passed on **time_sec**
KNOT_EVT_FLAG_LOWER_THRESHOLD	2	Send data every time that the item is below a threshold. The value to be compared with is the one passed on **lower_limit** (lower_int + lower_dec). If combined with **KNOT_EVT_FLAG_UPPER_THRESHOLD**, it is mandatory that **lower_limit** is smaller than **upper_limit**.

KNOT_EVT_FLAG_UPPER_THRESHOLD	4	Send data every time that the item is above a threshold. The value to be compared with is the one passed on **upper_limit** (upper_int + upper_dec). If combined with **KNOT_EVT_FLAG_LOWER_THRESHOLD**, it is mandatory that **lower_limit** is smaller than **upper_limit**.
KNOT_EVT_FLAG_CHANGE	8	Send data every time the item changes its value. Does not require any additional field.

Units and Values

```
#ifndef KNOT_TYPES_H
#define KNOT_TYPES_H

/*
 * This file defines the semantic for KNoT data types and unities
 * used by KNoT data sources.
 *
 */

// definition of invalid sensor id
#define KNOT_SENSOR_ID_INVALID               0xFF

// TypeIDs for basic units
#define KNOT_TYPE_ID_NONE                    0x0000
#define KNOT_TYPE_ID_VOLTAGE                 0x0001
#define KNOT_TYPE_ID_CURRENT                 0x0002
#define KNOT_TYPE_ID_RESISTENCE              0x0003
#define KNOT_TYPE_ID_POWER                   0x0004
#define KNOT_TYPE_ID_TEMPERATURE             0x0005
#define KNOT_TYPE_ID_RELATIVE_HUMIDITY       0x0006
#define KNOT_TYPE_ID_LUMINOSITY              0x0007
#define KNOT_TYPE_ID_TIME                    0x0008
#define KNOT_TYPE_ID_MASS                    0x0009
#define KNOT_TYPE_ID_PRESSURE                0x000A
#define KNOT_TYPE_ID_DISTANCE                0x000B
#define KNOT_TYPE_ID_ANGLE                   0x000C
#define KNOT_TYPE_ID_VOLUME                  0x000D
#define KNOT_TYPE_ID_AREA                    0x000E
#define KNOT_TYPE_ID_RAIN                    0x000F
#define KNOT_TYPE_ID_DENSITY                 0x0010
#define KNOT_TYPE_ID_LATITUDE                0x0011
#define KNOT_TYPE_ID_LONGITUDE               0x0012
#define KNOT_TYPE_ID_SPEED                   0x0013
#define KNOT_TYPE_ID_VOLUMEFLOW              0x0014
#define KNOT_TYPE_ID_ENERGY                  0x0015

// MAX TypeID for basic units
#define KNOT_TYPE_ID_BASIC_MAX
(KNOT_TYPE_ID_ENERGY+1)

// TypeIDs for logical units
#define KNOT_TYPE_ID_PRESENCE                0xFFF0
#define KNOT_TYPE_ID_SWITCH                  0xFFF1
#define KNOT_TYPE_ID_COMMAND                 0xFFF2

// MIN TypeID for logic units
#define KNOT_TYPE_ID_LOGIC_MIN
(KNOT_TYPE_ID_PRESENCE)
// MAX TypeID for logic units
```

```
#define KNOT_TYPE_ID_LOGIC_MAX
(KNOT_TYPE_ID_COMMAND+1)

#define KNOT_TYPE_ID_INVALID                  0xFFFF

// definition of measurement units
#define KNOT_UNIT_NOT_APPLICABLE              0x00
#define KNOT_UNIT_VOLTAGE_V                   0x01
#define KNOT_UNIT_VOLTAGE_MV                  0x02
#define KNOT_UNIT_VOLTAGE_KV                  0x03
#define KNOT_UNIT_CURRENT_A                   0x01
#define KNOT_UNIT_CURRENT_MA                  0x02
#define KNOT_UNIT_RESISTENCE_OHM              0x01
#define KNOT_UNIT_POWER_W                     0x01
#define KNOT_UNIT_POWER_KW                    0x02
#define KNOT_UNIT_POWER_MW                    0x03
#define KNOT_UNIT_TEMPERATURE_C               0x01
#define KNOT_UNIT_TEMPERATURE_F               0x02
#define KNOT_UNIT_TEMPERATURE_K               0x03
#define KNOT_UNIT_RELATIVE_HUMIDITY           0x01
#define KNOT_UNIT_LUMINOSITY_LM               0x01
#define KNOT_UNIT_LUMINOSITY_CD               0x02
#define KNOT_UNIT_LUMINOSITY_LX               0x03
#define KNOT_UNIT_TIME_S                      0x01
#define KNOT_UNIT_TIME_MS                     0x02
#define KNOT_UNIT_TIME_US                     0x03
#define KNOT_UNIT_MASS_KG                     0x01
#define KNOT_UNIT_MASS_G                      0x02
#define KNOT_UNIT_MASS_LB                     0x03
#define KNOT_UNIT_MASS_OZ                     0x04
#define KNOT_UNIT_PRESSURE_PA                 0x01
#define KNOT_UNIT_PRESSURE_PSI                0x02
#define KNOT_UNIT_PRESSURE_BAR                0x03
#define KNOT_UNIT_DISTANCE_M                  0x01
#define KNOT_UNIT_DISTANCE_CM                 0x02
#define KNOT_UNIT_DISTANCE_MI                 0x03
#define KNOT_UNIT_DISTANCE_IN                 0x04
#define KNOT_UNIT_ANGLE_RAD                   0x01
#define KNOT_UNIT_ANGLE_DEGREE                0x02
#define KNOT_UNIT_VOLUME_L                    0x01
#define KNOT_UNIT_VOLUME_ML                   0x02
#define KNOT_UNIT_VOLUME_FLOZ                 0x03
#define KNOT_UNIT_VOLUME_GAL                  0x04
#define KNOT_UNIT_AREA_M2                     0x01
#define KNOT_UNIT_AREA_HA                     0x02
#define KNOT_UNIT_AREA_AC                     0x03
#define KNOT_UNIT_RAIN_MM                     0x01
#define KNOT_UNIT_DENSITY_KGM3                0x01
#define KNOT_UNIT_LATITUDE_DEGREE             0x01
#define KNOT_UNIT_LONGITUDE_DEGREE            0x01
#define KNOT_UNIT_SPEED_MS                    0x01
#define KNOT_UNIT_SPEED_CMS                   0x02
```

```
#define KNOT_UNIT_SPEED_KMH                  0x03
#define KNOT_UNIT_SPEED_MIH                  0x04
#define KNOT_UNIT_VOLUMEFLOW_M3S             0x01
#define KNOT_UNIT_VOLUMEFLOW_SCMM            0x02
#define KNOT_UNIT_VOLUMEFLOW_LS              0x03
#define KNOT_UNIT_VOLUMEFLOW_LM              0x04
#define KNOT_UNIT_VOLUMEFLOW_FT3S            0x05
#define KNOT_UNIT_VOLUMEFLOW_GALM            0x06
#define KNOT_UNIT_ENERGY_J                   0x01
#define KNOT_UNIT_ENERGY_NM                  0x02
#define KNOT_UNIT_ENERGY_WH                  0x03
#define KNOT_UNIT_ENERGY_KWH                 0x04
#define KNOT_UNIT_ENERGY_CAL                 0x05
#define KNOT_UNIT_ENERGY_KCAL                0x06

// definition of value type
#define KNOT_VALUE_TYPE_INT                  0x01
#define KNOT_VALUE_TYPE_FLOAT                0x02
#define KNOT_VALUE_TYPE_BOOL                 0x03
#define KNOT_VALUE_TYPE_RAW                  0x04
#define KNOT_VALUE_TYPE_MIN                  KNOT_VALUE_TYPE_INT
#define KNOT_VALUE_TYPE_MAX
(KNOT_VALUE_TYPE_RAW+1)
#define KNOT_VALUE_TYPE_INVALID              0XFF
```