3. Разработка ПО контроллера STM32F303CBT6 для работы с уставками сигнализации.

Для работы с уставками синхронизации в ПО было внесено несколько существенных улучшений.

3.1 Дополнительные поля в структуре, хранящей настройки.

struct SettingsMeasures

{

int limit\_min[NUM\_MEASURES\_TO\_CONTROL];

int limit\_max[NUM\_MEASURES\_TO\_CONTROL];

},

где:

* limit\_min – минимальное допустимое значение измеряемой величины, при достижении которого сработает звуковая и световая сигнализация:
* limit\_max – максимальное допустимое значение измеряемой величины, при достижении которого сработает звуковая и световая сигнализация.

3.2. Дополнительные пункты в меню настроек.

Например, для температуры так:

#define COMMON\_PART\_ITEM TypeItem::E type; \

pchar title; \

const Page \*keeper; \

FuncOpenClose funcOnOpenClose; \

FuncAfterDraw funcOnDraw

struct DGovernor

{

COMMON\_PART\_ITEM;

int min;

int max;

int \*value;

};

#define DEF\_GOVERNOR(name, title, keeper, beforeOpen, funcAfterDraw, min, max, value) \

static const DGovernor name = { TypeItem::Governor, title, (const Page \*)&keeper, beforeOpen, funcAfterDraw, min, max, &(value) };

#define DEF\_GOVERNOR\_MIN(\_name, page\_self, \_min, \_max, type)

DEF\_GOVERNOR(\_name, "Предел мин", \*page\_self, nullptr, nullptr, \_min, \_max, gset.measures.limit\_min[type])

DEF\_GOVERNOR\_MIN(gTemperatureLimitMin, PageMeasures::Temperature::self, -30, 60, Measure::Temperature);

3.3 Проверка на выход за пределы допустимого диапазона в каждом цикле измерений.

bool Measure::InRange() const

**{**

if (((int)name) >= 0 && name < NUM\_MEASURES\_TO\_CONTROL)

{

if (GetDouble() < gset.measures.\_limit\_min[name])

{

return false;

}

if (GetDouble() > gset.measures.limit\_max[name])

{

return false;

}

}

return true;

**}**

3.4 Собственно, сигнализация, если обнаружен выход из диапазона.

Световая:

String<>("%s", measure.Name().c\_str()).Draw(x0, y, measure.value.InRange() ? Color::WHITE : Color::FLASH\_10);

Звуковая:

void Beeper::Update()

{

Storage::AllLastMeasuresInRange() ? Stop() : Start(4000);

need\_sound ? Start(4000) : Stop();

}