



Рис. 2 Пример задания Web-квеста

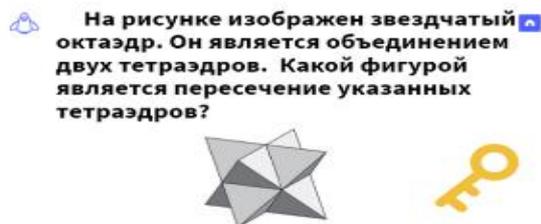


Рис. 3 Пример задания Web-квеста

Знакомясь с материалами для изучения перед выполнением задания, обучающийся также работает с моделями многогранников. Для выполнения задания, представленного на Рис. 4, участникам предлагается изготовить модель многогранника наиболее удобным для них способом, используя развертку многогранников, оригами или другие способы построения реальной модели многогранника. При самостоятельном изготовлении модели образ создается по частям, что позволяет «ощутить» абстрактные представления о звездчатых многогранниках.

Заключение. Таким образом, использование математических моделей является неотъемлемой частью современных исследований, в том числе при изучении звездчатых многогранников. Модели многогранников используются в различных видах: графическом, математическом, описательном. Используя дистанционные технологии и современные информационные ресурсы можно сделать процесс изучения звездчатых многогранников более интересным и привлекательным для обучающихся. У них появляется возможность изучить этот материал в посильном для себя режиме, в условиях онлайн обучения. Познакомиться с материалами разработанного квеста можно по ссылке: <https://sites.google.com/view/cowboywildwest>.



Рис. 4 Пример задания Web-квеста

1. Веннинджер, М. Модели многогранников/ М. Веннинджер. – Москва: «Мир»,1974. – 236 с.
2. Приказ Минобрнауки России от 17.05.2012 N 413 (ред. от 11.12.2020) "Об утверждении федерального государственного образовательного стандарта среднего общего образования" (Зарегистрировано в Минюсте России 07.06.2012 N 24480) {КонсультантПлюс} – URL: http://www.consultant.ru/document/cons_doc_LAW_131131/ (Дата обращения: 08.09.2021)

ПРИНЦИПЫ РАЗРАБОТКИ ПРОГРАММНЫХ АБСТРАКЦИЙ ДЛЯ РАДИОМОДУЛЕЙ LORA

Довгулевич Д.А.,

*молодой ученый ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь
Научный руководитель – Кашевич И.Ф., канд. физ.-мат наук, доцент*

В 21 веке начала активное развитие концепция, называемая Интернет Вещей (Internet of Things) или сокращенно IoT.

Это концепция, суть которой заключается в объединении предметов в единую сеть для взаимодействия между собой и внешним миром. Эта концепция активно развивается

и внедряется как в быту, для автоматизации рутинной работы, так и в промышленности. Для взаимодействия вещей друг с другом традиционные интернет сети, такие как Wi-Fi, GSM, Ethernet, оказались плохо пригодны из-за высокого потребления электроэнергии конечных модемов и небольшой емкости сети.

Поэтому для IoT были разработаны свои стандарты сетей передачи данных, такие как BLE, ZigBee, NB-IoT, LoRa, LTE-M и ряд других. Среди всех стандартов, разработанных для IoT, можно выделить отдельную группу – Low-power Widearea Network (энергоэффективные сети дальнего радиуса действия) или сокращенно LPWAN. Одним из представителей таких стандартов является LoRa.

Для разработки устройств, использующих этот способ связи, необходимо использовать соответствующие радиомодули.

Целью исследования являлось описание принципов разработки слоя программных абстракций, упрощающего процесс разработки устройств за счет абстрагирования от написания кода на уровне регистров, и избавления от привязки программного обеспечения к конкретному микроконтроллеру.

Материал и методы. Материалом для исследования послужило исходное программное обеспечение, предоставляемое производителями трансиверов, а также официальная документация на радиомодуль SX127x. При проведении исследований применялись общенаучные методы исследования, и процедурные методы разработки программного обеспечения.

Результаты и их обсуждение. Для работы с радио модулями LoRa работающими на трансиверах серии SX127x необходимо управлять регистрами с помощью интерфейса SPI. Для удобства работы лучше создать слой программных абстракций, позволяющий вызывать функции в место непосредственного управления регистрами. Для этого необходимо написать соответствующий драйвер. Назначение и адреса регистров можно узнать из документации на трансивер. Производитель предоставляет сводную таблицу по регистрам, которая описывает адреса, название, назначение и значение по умолчанию всех регистров трансивера. также для каждого регистра в документации есть детальное описание. Большинство регистров управляется побитово, и группы битов отвечают за различные параметры (таблица 1).

Таблица 1 – описание регистра RegPaConfig

RegPaConfig (0x09)	7	PaSelect	rw	0x00	Selects PA output pin 0 → RFO pin. Maximum power of +14 dBm 1 → PA_BOOST pin. Maximum power of +20 dBm
	6-4	MaxPower	rw	0x04	Select max output power: $P_{max}=10.8+0.6*MaxPower$ [dBm]
	3-0	OutputPower	rw	0x0f	$P_{out}=P_{max}-(15-OutputPower)$ if PaSelect = 0 (RFO pins) $P_{out}=17-(15-OutputPower)$ if PaSelect = 1 (PA_BOOST pin)

Для удобной работы с этими регистрами необходимо составить список обозначений, связывающий адреса регистров с их названиями. Данное действие позволит обращаться к регистрам не по адресам, а по названиям, что избавляет от необходимости запоминать адреса или, постоянно сверяться с документацией.

Такой список будет выглядеть следующим образом:

```
#define LR_RegFifo                                0x00
//                                               Common settings
#define LR_RegOpMode                              0x01
#define LR_RegFrMsb                               0x06
#define LR_RegFrMid                               0x07
#define LR_RegFrLsb                               0x08
//                                               Tx settings
#define LR_RegPaConfig                            0x09
```

```

#define LR_RegPaRamp 0x0A
#define LR_RegOcp 0x0B
// Rx settings
#define LR_RegLna 0x0C
...

```

В данном списке должны быть описаны все регистры с их адресами даже те, которые использовать в данный момент не планируется. Это упростит работу в будущем и избавит от необходимости дополнять список каждый раз при введении новой функции. Аналогичным образом необходимо составить список значений, которые будут записываться в эти регистры.

Далее необходимо реализовать базовые функции работы с трансивером, такие как чтение/запись регистров, задержки, отправка команд, инициализация, перезагрузка.

На пример для работы с SPI нам необходим управляющий вывод NSS отвечающий за выбор чипа. Для отправки данных вывод NSS должен находится в логическом нуле, иначе чип не будет обрабатывать передаваемые данные.

Для управления этим выводом с помощью контроллера STM32F103 была реализована функция *SX127X_hw_SetNSS*, она имеет следующий вид:

```

__weak void SX127X_hw_SetNSS(SX127X_hw_t * hw, int value) {
    HAL_GPIO_WritePin(hw->nss.port, hw->nss.pin,
                      (value == 1) ? GPIO_PIN_SET:GPIO_PIN_RESET);
}

```

На вход функция принимает структуру, содержащую номер вывода и название порта, а также значение, которое необходимо установить. Внутри она вызывает функцию HAL_GPIO_WritePin являющуюся функцией библиотеки HAL, она устанавливает соответствующий пин микроконтроллера в необходимый уровень (логический 0 или логическая 1).

Функция объявляется с ключевым словом `__weak`, которое означает, что данное определение функции будет использоваться до тех пор, пока не найдется хотя бы одно совпадающее по написанию без этого ключевого слова. Такое объявление позволяет легко перегружать функции, которые привязаны к микроконтроллеру, не изменяя сам драйвер. Аналогичным образом описываются все необходимые для работы с радиомодулем функции.

Заключение. Для удобства работы контроллера с внешними устройствами рекомендуется реализовать слой программных абстракций, который позволяет ускорить процесс разработки программного обеспечения и позволит абстрагироваться от управляющего микроконтроллера. Данная абстракция должна описывать все используемые регистры. Так же необходимо создать структуры со значениями которые могут принимать эти регистры и описать функции для работы с контроллером. Данный принцип разработки применим не только к конкретному радио модулю, но может быть использован для любых других периферийных устройств, подключаемых к микроконтроллерам.

1. AN1200.22 LoRa™ Modulation Basics [Электронный ресурс] // Revision 2, May 2015 Semtech Corporation // Режим доступа: <https://www.mouser.com/datasheet/2/761/sx1276-1278113.pdf> – Дата доступа: 28.08.2020.

2. Application Note: MCU Requirements for LoRaWAN [Электронный ресурс] // AN1200.28 Rev 3 December 2017 // Режим доступа: <https://semtech.my.salesforce.com/sfc/p/#E0000000elG/a/2R000000HSRS/> – Дата доступа: 28.08.2020.