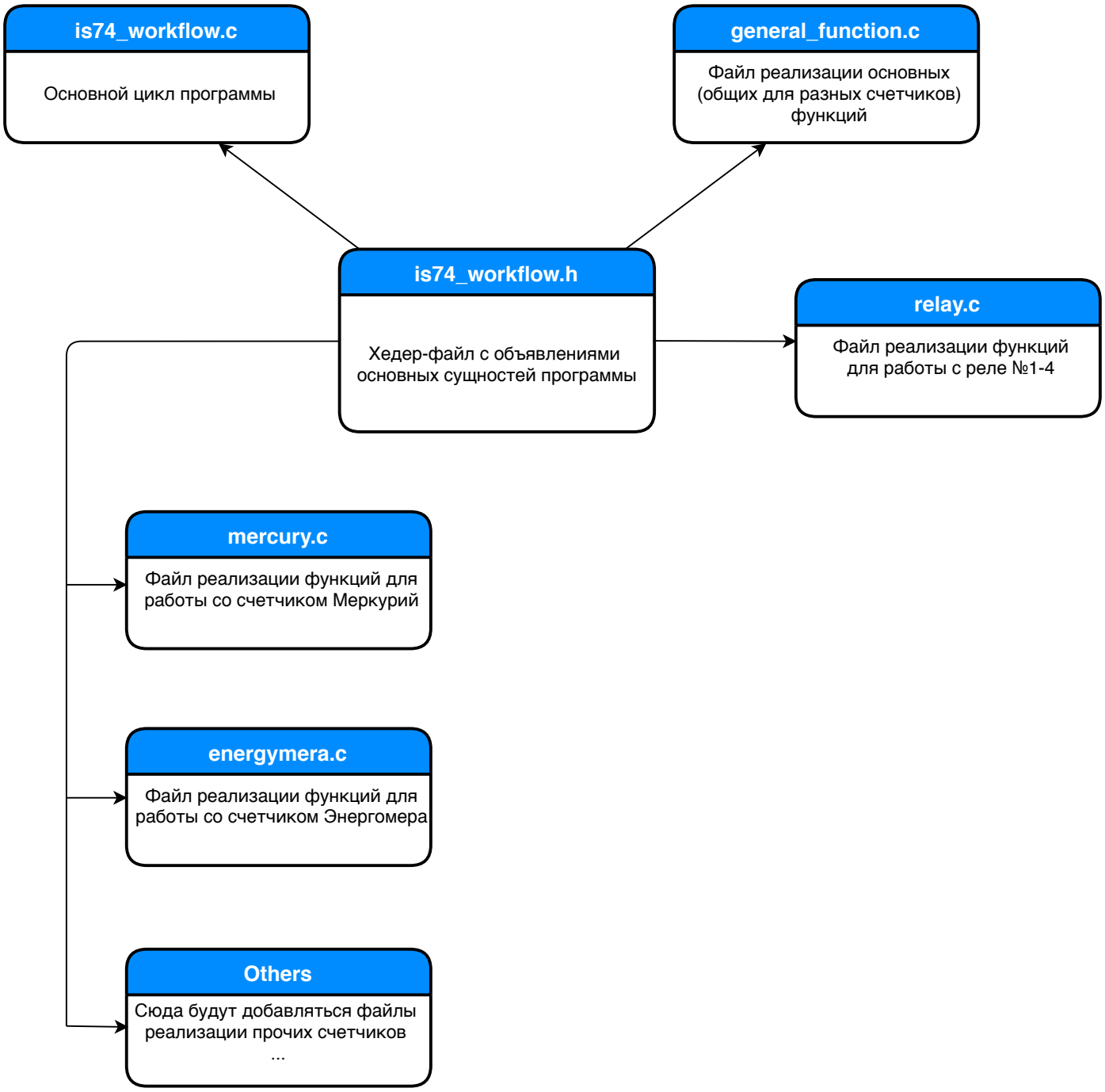
1. Структура проекта **Is74_LoRa_Module** (Электросчетчики)



2. Описание проекта **IS74_LoRa_Module** (Электросчетчики)

1. **is74_workflow.h** - Хедер-файл с объявлениями основных сущностей программы.

1. Перечисление **__is74_mainstate** - Реализует модель конечно автомата со следующими возможными состояниями:

1.1 **IS74_STATE_IDLE** - Не используется;

1.2 **IS74_STATE_WAIT_FOR_RESPONSE** - Основное состояние, при котором программа работает в штатном режиме. Так же опрашивает необходимые счетчики, обрабатывает команды пришедшие от LoRa-сервера и так далее;

1.3 **IS74_STATE_WAIT_FOR_JOIN** - Состояние при котором программа ожидает физическое подключение модуля к сети:
в случае неудачи переходит в состояние IS74_STATE_NO_LORAWAN_NET;
в случае успеха запрашивает реальное время у сервера и переходит в состояние IS74_STATE_RS485_CONNECT;

1.4 **IS74_STATE_RS485_CONNECT** - Состояние при котором программа подключается к таргетному (указанному в переменной whoami) счетчику;

1.5. **IS74_STATE_NO_LORAWAN_NET** - Состояние автомата, когда модуль не смог подключиться к сети LoRa. Попав в данное состояние программа вызывает функцию is74__OneTime() (смотри назначение и описание данной функции в файле general_function.c)

2. Структура **Connections** - Содержит переменные соединения + некоторые другие:

2.1 **whoami** - Переменная производителя таргетного счетчика. (Энергомера = 1, Меркурий = 2 и т.д.);

2.2 **netAddr** - Сетевой адрес текущего счетчика. Используется при запросе с кодом 05. (смотри описание запросов);

2.3 **desc_integral** - Тип считываемых данных. (Например 81 = помесячные интегральные данные, 84 - текущие данные,...). Используется при запросе с кодом 05. (смотри описание запросов);

2.3 **netAddrDev[6]** - Количество и список адресов счетчиков, которые опрашивает программа:
netAddrDev[0] - количество адресов счетчиков. Max = 5;
netAddrDev[1] - адрес 1-го счетчика для считывания;
netAddrDev[2] - адрес 2-го счетчика для считывания и т. д...;

2.4 **request_04** - Флаговая переменная запроса с кодом 04. (смотри описание запросов):
Если переменная равна true значит пришел запрос с кодом 04 и необходимо его обработать;

2.5 **request_05** - Флаговая переменная запроса с кодом 05. (смотри описание запросов):
Аналогично переменной request_04;

2.6 **l_day[5]** - Список последних отправленных дней (Интегральные дневные):
l_day[0] - последний день для 1-го счетчика;
l_day[1] - последний день для 2-го счетчика если есть и т. д...;

2.7 **l_month[5]** - Аналогичен списку в пункте 2.6 только для месяца;

2.8 **send_day** - Флаговая переменная обновления дневных интегральных данных в счетчике:
Если переменная равна true значит сменился день в счетчике и необходимо отправить новые данные в текущей сессии отправки;

2.9 **send_month** - Аналогична переменной в пункте 2.8 только для месяца;

2.10 **isOpen** - Флаговая переменная соединения со счетчиком по UART. Равна:
1 - если удалось соединиться с счетчиком для считывания метрик;
0 - если не удалось соединиться с счетчиком для считывания метрик;

2.11 **index_com** - Используется для последовательного считывания метрик;

2.12 **is_send** - Переменная определяет действие:
0 - считываем метрики с текущего счетчика;
1 - отправляем считанные метрики на LoRa-сервер;

2.13 **complete** - Не помню зачем, кек;

2.14 **delay** - Определяет задержку между отправками считанных текущих данных на LoRa-сервер;

2.15 **t** - Время начала сессии опроса счетчика. (Необходима для решения проблемы смещения времени отправки метрик на LoRa-сервер);

2.16 **delta_t** - Время затраченное на опрос счетчика в sec. (Необходима для решения проблемы смещения времени отправки метрик на LoRa-сервер);

2.17 **error** - Ошибка при конекте с счетчиком и последующем считывании метрик:
1 - Не удалось подключиться к счетчику;
2 - Какая-то метрика не считалась;

2.18 **delta_delay** - Меняем задержку с учетом затраченного времени на считывание. delay - delta_t * 1000. (Необходима для решения проблемы смещения времени отправки метрик на LoRa-сервер);

2.19 **flag_relay_status**, **relay[4]** - переменные для работы с реле №1-4;

3. Структура **RelayStrats** - Содержит переменные для работы с реле №1-4;

4. Перечисление **__is74_whoami** - Список идентификаторов производителей счетчиков;

5. Перечисление **__is74_UART_Init_Variable** - Конфигурация ком-порта;

6. Структуры описывающие конкретный счетчик + команды описанные в конкретном протоколе;

7. Раздел **Function prototype** - объявление всевозможных функций;

2. **is74_workflow**.c - Основной цикл программы. (Думаю, что комментарии излишни).

3. **general_function.c** - Файл реализации основных (общих для разных счетчиков) функций.

Описание функций:

1. **void is74__UART_Init(void)**:
Инициализирует настройку ком-порта;

2. **void is74__OneTime(void)**:
Вызывается перед работой основного цикла программы.
Инициализирует стартовое состояние конечного автомата;

3. **void setup_config(void)**:
Стартовая инициализация основных переменных программы.
Считывание значений из EEPROM (если такие имеются);

4. **void is74_PrepareTx(void)**:
Не используется (нет реализации);

5. **uint16_t Crc16MudBus(uint8_t *msg, uint8_t len)**:
Возвращает 16 битную контрольную сумму (ModBus). На вход принимает 2 аргумента:
1-ый - Указатель на массив для которого вычисляется CRC;
2-ой - Длина массива для которого вычисляется CRC;

6. **uint16_t Crc8Calc(uint8_t *msg, uint8_t len)**:

Возвращает 8 битную контрольную сумму. На вход принимает 2 аргумента:

1-ый - Указатель на массив для которого вычисляется CRC;

2-ой - Длина массива для которого вычисляется CRC;

7. **void clear_ans(void)**:

Отчищает массивы запросов/ответов (UART2_TX_buff[]/UART2_RX_buff[]);

8. **void is74_delay(uint8_t d_enr, uint8_t d_mer)**:

Реализует задержку между запросами к счетчикам;

1-ый - Задержка для Энергомеры = DEL_COM_ENR;

2-ой - Задержка для Меркурия = DEL_COM_MER;

9. **void wrapper_crc(uint8_t *arr, uint8_t size)**:

Записывает 16 битную контрольную сумму (для массива в 1-м аргументе)
в 2 последних элемента того же массива, 2-ой аргумент соответственно размер;

10. **uint32_t date_to_epoch(uint8_t *arr, uint8_t i)**:

Конвертирует формат даты удобной для человека в Unix-time
Возвращает Unix-time

11. **struct tm *epoche_to_date()**:

Обратная для функции в пункте 10;
Возвращает стандартную структуру tm (где год, месяц, день, час,...)

12. **uint8_t is_empty_metrika()**:

Возвращает 0 если был ответ при считывании метрики;
Возвращает 2 если не было ответа при считывании метрики;

13. **void request_handler(uint8_t *buf, uint8_t len)**:

Функция-обработчик команд (запросов) от LoRa-сервера. В ответ присылает сообщение на LoRa-сервер
(описание присылаемых сообщений и кодов запроса смотри в файле "Описание запросов") Принимает 2 аргумента:

1-ый массив запроса, где buf[0] - код запроса;

2-ой размер массива запроса;

4. Файл **mercury.c** - Файл реализации функций для работы со счетчиком Меркурий.

Описание функций:

1. **void connect_mercury(void)**:

Функция пытается подключиться к счетчику 10 раз:
В случае неудачи error = 1;

2. **void init_mercury(void)**:

Инициализирует служебную информацию считанную из счетчика и записывает
серийный номер в EEPROM;

3. **void init_time_date_mercury(void)**:

Инициализирует дату и время считанную из счетчика, а так же проверяет
изменения суток, или месяца (в случае их изменения необходимо отправить на LoRa-сервер).
Так же записывает их в EEPROM;

4. **void request_mercury(uint8_t *req, uint8_t len, uint16_t size_rec)**:

Формирует запрос к счетчику. Принимает 3 аргумента:

1-ый - определенная команда в соответствии с протоколом;

2-ой - размер команды выше;

3-ий - количество байт ожидаемых в ответе;

5. **void set_net_addr(uint8_t addr)**:

Записывает в команду адрес текущего счетчика;

6. **uint16_t addr_return(uint8_t arg)**:

Возвращает адрес;

7. **void init_propow_mercury(void)**:

Инициализирует конфигурацию профиля мощности, записывает в команду считывания профиля мощности адрес последней даты записи;

8. **void run_mercury(void)**:

Последовательно считывает все необходимые метрики;

9. **void send_lora_mercury(void)**:

Если error = 0 отправляет на LoRa-сервер считанные метрики;

Если error = 1 отправляет на LoRa-сервер ошибку "Нет ответа от счетчика";

Если error = 2 отправляет на LoRa-сервер ошибку "Неполные данные";