Методические указания финального кейса НТО 2022-2023

Сенсорная часть проекта

Необходимо реализовать потоковую передачу данных, снятых с сенсорных модулей. Задача состоит из трех частей:

- Снятие данных с АЦП
- Обработка данных и отправка их на сервер
- Прием управляющего сигнала с сервера (для внутреннего сенсорного и исполнительного модуля)

Аналогово-цифровое преобразование (АЦП)

Под термином «сигнал» [1] мы традиционно понимаем некоторый физический процесс (например, изменяющееся во времени напряжение), отображающий некоторую информацию. Математически сигнал описывается функцией *f(t)* *о*пределенного вида. Характеризовать виды сигналов будем с использованием следующих понятий.

Аналоговый сигнал – описывается непрерывной (или кусочно-непрерывной) функцией *f(t)*.

Дискретный сигнал – это функция дискретного аргумента $y = y(nT) = y(n) = \{y_n\}$ с областью определения $D = \{nT \vee n \in Z\}$. Значения $y(nT)$ называются отсчетами сигнала, а величина *T*, представляющая собой расстояние по оси аргумента между соседними отсчетами, называется периодом (шагом) дискретизации.

Цифровым сигналом называется дискретный сигнал, который может принимать значения из конечного множества чисел, $\forall n\, y(n) \in \{d_1, \ldots, d_M\}$. Возможные значения d_j *(j=1,...,M)* называются уровнями сигнала.

Сопоставление функции непрерывной переменной функцию дискретной переменной называется дискретизацией.

Так как дискретная функция имеет непрерывную ОДЗ, записать значение в память цифровой аппаратуры не представляется возможным, так как для этого потребовалось бы бесконечно много памяти. В связи с этим, требуется разбиение диапазона оценки входной функции на уровни – кванты. Такой процесс называется квантованием уровней.

Техническая система, которая может преобразовывать входной аналоговый сигнал в соответствующий ему цифровой называется аналогово-цифровым преобразователем.

В esp32 есть два многоканальных 12-битных АЦП. Вам достаточно выбрать один канал и настроить на него АЦП. Чтение значения на входе может быть выполнено функцией adc1_get_raw(). Перед вызовом этой функции, необходимо сконфигурировать АЦП. С примером конфигурации можно ознакомиться на официальном сайте espressif [2]. В среде Arduino IDE конфигурации не требуется.

В ESP32 драйвер АЦП предоставляет девять функций:

1. analogRead(pin): получить значение АЦП для указанного вывода.
2. analogReadResolution(bits): установите разрешение вывода аналогового чтения. По умолчанию 12-битный, но возможные значения от 9 до 12.
3. analogSetWidth(bits): устанавливает биты выборки и разрешение чтения. По умолчанию 12-битный, но диапазон от 9 до 12.
4. analogSetClockDiv(clockDiv): установите делитель для часов АЦП.

5. analogSetAttenuation(аттенюация): установите затухание для всех каналов. По умолчанию 11 дБ, но возможные значения: 0 дБ, 2,5 дБ, 6 дБ и 11 дБ.
6. analogSetPinAttenuation(pin, затухание): установите затухание для определенного вывода.

7. adcAttachPin(pin): прикрепить вывод к АЦП (выполняется автоматически в AnalogRead).
8. analogSetVRefPin(pin): установите вывод, который будет использоваться для калибровки АЦП, если ESP32 еще не откалиброван. Возможные контакты: 25, 26 или 27.
9. analogReadMilliVolts(pin): получить значение в милливольтах для вывода.

Помните, что АЦП 12- битный, поэтому желательно обнулить старшие биты результата вызова функции (там может содержаться мусор). Для этого необходимо наложить битовую маску: res &= 0x0FFF;

Для работы с датчиком LM235Z, Вы можете ознакомиться с соответствующим информационным листом [3].

Отправка данных на сервер

Отправка данных на сервер осуществляется по средствам tcp-клиента.

При снятии отсчётов с АЦП, их необходимо располагать в памяти один за другим, без каких-либо разрывов — так вы сможете отправлять данные одним вызовом send. Этого можно достичь, используя массивы.

Кроме того, поскольку АЦП 12-битный, каждый отсчёт будет занимать два байта памяти, а значит вам необходимо позаботиться о порядке байт.

Может показаться, что поскольку передача будет потоковая, вам больше подойдёт UDP, однако лучше воздержаться от использования этого протокола, поскольку вам придётся позаботиться о порядке пришедших пакетов и их размере.

За основу проекта лучше взять пример tcp-клиента в esp-idf. В этом случае, у вас уже будут реализована функция подключения к wifi (example_connect) [4]. Настройка данных wifi-сети выполняется через менюконфиг (вызывается командой idf.py menuconfig). В подменю Example Connection Configuration. Кроме того, в подменю Example Configuration можно указать ip-адрес и порт сервера (однако возможно это сделать из файла с исходниками, в то время как настройки wifi в исходниках будет указать сложнее).

Серверная часть проекта

Серверу необходимо принять соединение сенсорного модуля, и начать считывать отправляемые данные (позаботившись о порядке следования байт). После приема и обработки данных с сенсорных датчиков необходимо произвести расчет температуры формирования и дальнейшей отправки управляющего сигнала. Алгоритм, который необходимо реализовать прописан в условии задачи. Отправлять управляющий сигнал клиенту (внутренний сенсорный и исполнительный модуль) рекомендуется массивом для успешной реализации дифференциальной скорости работы вентилятора.

Широтно-импульсная модуляция (ШИМ)

Широтно-импульсная модуляция (ШИМ) или Pulse-Width Modulation — это метод управления средней мощностью, подаваемой электрическим сигналом на нагрузку. Среднее значение напряжения (и тока), подаваемого на нагрузку, регулируется поочередным скачкообразным переключением питания от 0 до 100 % от опорного напряжения со скоростью, превышающей скорость значительного изменения нагрузки. Чем дольше переключатель включен, тем выше общая мощность, подаваемая на нагрузку.

ШИМ особенно подходит для работы с инерционными нагрузками, такими как двигатели, на которые не так легко влияет это дискретное переключение. Цель ШИМ — управлять нагрузкой; тем не менее, частота переключения ШИМ должна быть тщательно выбрана, чтобы сделать это плавно.

Частота переключения ШИМ может сильно различаться в зависимости от нагрузки и приложения. Например, в электрической плите переключение должно производиться всего несколько раз в минуту; 100 или 120 Гц (удвоенная частота бытовой электрической сети) в диммере; от нескольких килогерц (кГц) до десятков кГц для электродвигателя; и до десятков или сотен кГц в аудиоусилителях и блоках питания компьютеров. Выбор слишком высокой частоты коммутации приводит к плавному управлению нагрузкой, но может привести к преждевременному выходу из строя механических компонентов управления. Выбор частоты коммутации, слишком низкой для приложения, вызывает колебания нагрузки, что также способствует преждевременному износу.

Основное преимущество ШИМ заключается в том, что потери мощности в коммутационных устройствах очень малы. При выключенном выключателе тока практически нет, а при включенном и передаче питания на нагрузку практически нет падения напряжения на

ключе. Таким образом, потери мощности, являющиеся произведением напряжения и тока, в обоих случаях близки к нулю.

$$R_{tr} \to \infty \leftrightarrow P = \frac{U^2}{R} \to 0,$$
$$R_{tr} \to 0 \leftrightarrow P = I^2 R \to 0.$$

ШИМ также используется в некоторых системах связи, в которых его рабочий цикл использовался для передачи информации по каналу связи.

В электронике многие современные микроконтроллеры интегрируют ШИМ-контроллеры, открытые для внешних контактов, в качестве периферийных устройств, управляемых прошивкой с помощью внутренних интерфейсов программирования. Они обычно используются для управления двигателем постоянного тока в робототехнике, регулировании импульсного источника питания и других приложениях.

В широтно-импульсной модуляции используется прямоугольная импульсная волна, с регулируемой шириной импульса, генерируемая на заданной частоте. Изменение длительности импульса приводит к изменению среднего значения формы волны. Длина импульса устанавливается с помощью специального коэффициента – рабочего цикла.

Термин «рабочий цикл» описывает пропорцию «включенного» времени к регулярному интервалу или «периоду» времени; низкий рабочий цикл соответствует малой мощности, поскольку большую часть времени питание отключено. Рабочий цикл выражается в процентах, 100 % — полностью включено. Когда цифровой сигнал включен половину времени и выключен другую половину времени, цифровой сигнал имеет рабочий цикл 50% и напоминает «прямоугольную» волну. Когда цифровой сигнал находится во включенном состоянии больше времени, чем в выключенном, его

коэффициент заполнения >50%. Когда цифровой сигнал находится в выключенном состоянии больше времени, чем во включенном, его рабочий цикл составляет <50%. Приведем рисунок, иллюстрирующий понятие рабочего цикла ШИМ (Рисунок 1).

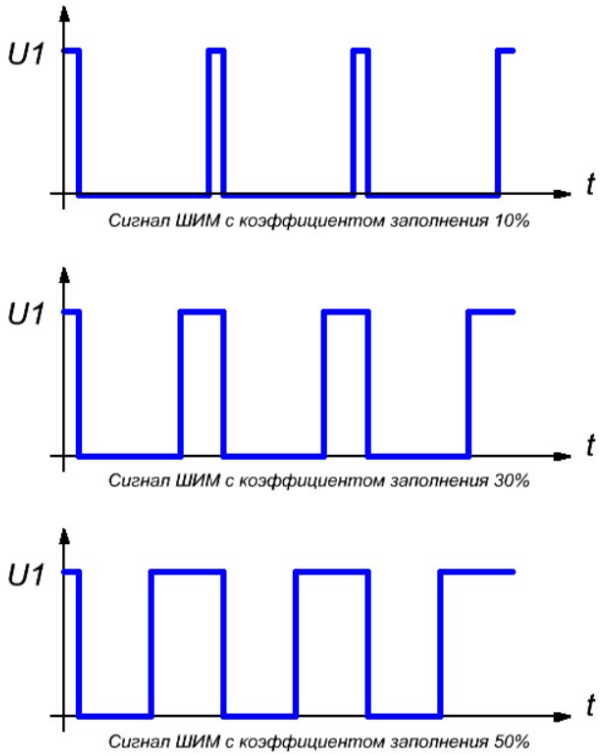


Рисунок 1

Рассмотрим форму импульса *f(t)*, с периодом *T*, низкое значение y_{min}, высокое значение y_{max}, рабочий цикл *D* (см. рисунок 2), среднее значение формы волны определяется как:

$$\bar{y} = \frac{1}{T}\int_0^T f(t)\,dt$$

$$\begin{aligned}\bar{y} &= \frac{1}{T}\left(\int_0^{DT} y_{\max}\,dt + \int_{DT}^{T} y_{\min}\,dt\right)\\ &= \frac{1}{T}\left(D\cdot T\cdot y_{\max} + T\left(1-D\right)y_{\min}\right)\\ &= D\cdot y_{\max} + \left(1-D\right)y_{\min}\end{aligned}$$

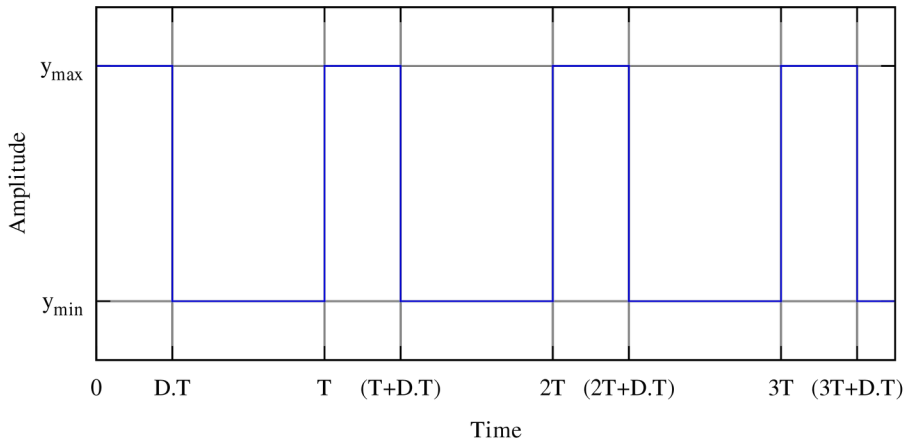


Рисунок 2

Последнее выражение может быть довольно упрощено во многих случаях: если положить y_{min} = 0 , то $\overline{y} = D \cdot y_{max}$. Отсюда среднее значение сигнала напрямую зависит от коэффициента заполнения *D*.

Простейший способ генерации ШИМ-сигнала — аналоговый вариант (рисунок 3), при котором несущая, представленная «треугольником» или «пилой», поступает на инвертирующий входной узел компаратора. Основной обрабатываемый сигнал подается на его «прямой» вход. Если значение несущей в данный момент превышает по амплитуде полезный сигнал, то на выходе компаратора формируется нулевой уровень. В ситуации, когда его величина меньше сравниваемого – на том же выходе появляется отрицательная «единица» (низкий уровень).

В итоге, посредством компаратора, формируется дискретный сигнал, имеющий определенную частоту. Последняя соответствует периодичности входной «пилы» или «треугольника». А длина импульсной посылки на выходе устройства пропорциональна уровню основного или рабочего напряжения. То есть, широтно-импульсная модуляция позволяет получить соответствующее представление аналогового сигнала, существенно облегчающее его обработку и упрощающее электронную схему.

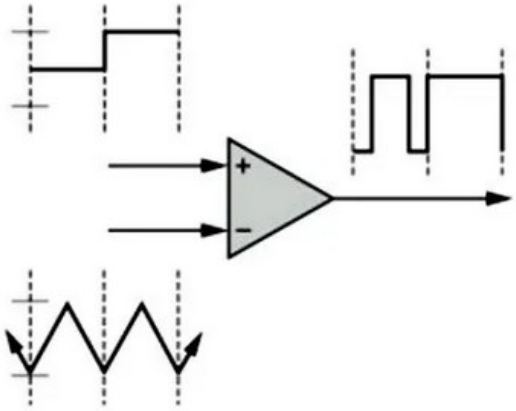


Рисунок 3

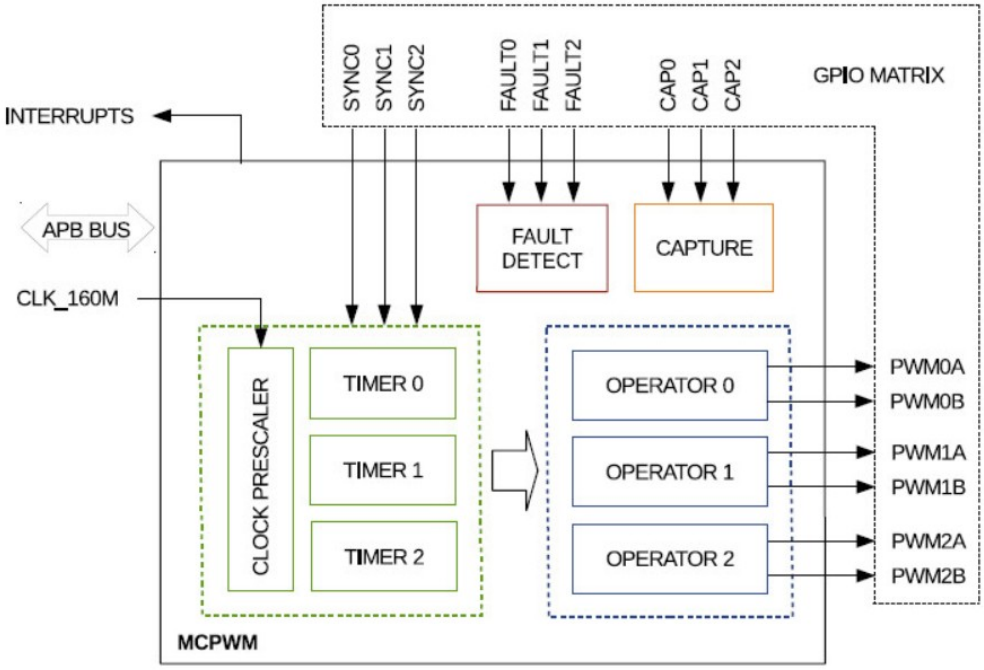
Если же пилообразный «опорный» сигнал поступает на плюсовой или «прямой» вход компаратора, а основной – на инвертирующий, то прямоугольные импульсы на выходе будут положительными.

Более подробно о схемах, применяемых для генерации ШИМ-сигнала Вы можете узнать в источниках [7-9].

В официальной документации espressif, содержащей информацию об API для МК ESP32, способы ШИМ-контроля разделены на 2 подпункта:

- управление двигателями [5],
- LED источники света [6].

ESP32 имеет два модуля, которые управляют шириной импульсов в ШИМ. Каждый из них имеет три блока выходов (Рисунок 4).



MCPWM Module Overview

Рисунок 4

Разделение функций для управления светодиодами и электромоторами достаточно условно, и всё может быть взаимозаменяемым: LED-контроль можно использовать для управления двигателями и наоборот. Однако для двигателей там имеется специализированный API, предназначенный для использования из-под среды ESP-IDF.

На практике чаще всего задействуется именно управление с помощью LED-подхода, в рамках которого система предоставляет нам 16 независимых каналов, каждый из которых может быть настроен независимо от других.

Все 16 каналов тактируются генератором шины APB с частотой 80 МГц, 8 из которых имеет возможность выбрать тактовую частоту в 8 МГц. Эта функция предназначена для создания энергоэффективных систем с малым потреблением.

В таблице ниже приводится официальная информация о наиболее частых конфигурациях выходного ШИМ сигнала:

Таблица 1

Commonly-used Frequencies and Resolutions

LEDC Clock Source	LEDC Output (PWM) Frequency	Highest Resolution
APB_CLK (80 MHz)	1 kHz	1/65536 (16 bit)
APB_CLK (80 MHz)	5 kHz	1/8192 (13 bit)
APB_CLK (80 MHz)	10 kHz	1/4096 (12 bit)
RTC8M_CLK (8 MHz)	1 kHz	1/4096 (12 bit)
RTC8M_CLK (8 MHz)	8 kHz	1/512 (9 bit)
REF_TICK (1 MHz)	1 kHz	1/512 (9 bit)

Так как ESP32 не имеет возможности управлять ШИМом с помощью Arduino-функции analogWrite(), то допустимо применять библиотеку esp32-gal-ledc.h, которая предоставляет следующие функции для управления ШИМом:

- ledcSetup(канал, частота, разрешение);
- ledcAttachPin(пин, канал);
- ledcWrite(канал, коэффициент заполнения);
- ledcRead(канал);
- ledcWriteTone(канал, частота);
- ledcReadFreq(канал);
- ledcDetachPin(пин);

Более подробно о работе сервоприводов Вы можете узнать в соответствующем источнике [10].

Для корректной работы с сервоприводом TowerPro MG995 требуется соответствующая конфигурация ШИМ. Вы, также, можете ознакомиться с информационным листом сервопривода по ссылке [11].

Чтобы двигатель постоянного тока начал вращаться, ему необходимо обеспечить нужное количество энергии. Как правило, для маломощных двигателей достаточно несколько ватт. Блок управления (микроконтроллер), который принимает решения о запуске двигателя, не может непосредственно управлять двигателем, то есть обеспечить необходимую мощность со своего вывода. Это связано с тем, что

порты микроконтроллера имеют очень ограниченную нагрузочную способность (максимальный ток на выходе микроконтроллера обычно не более 20 мА). Поэтому нужен усилитель мощности — устройство, которое может на своем выходе генерировать сигнал мощностью большей, чем мощность на его входе. Такими устройствами являются транзистор и реле, которые прекрасно подходят для управления двигателем постоянного тока.

Самым простым способом приведения в действие электродвигателя является применение биполярного транзистора, используемого в качестве переключателя. Однако, необходимо в схему добавить резистор R, подобрав его таким, чтобы в худшем случае (когда потенциал базы равен потенциалу эмиттера) через него протекал ток, не превышающий максимальный ток порта микроконтроллера. Поскольку двигатель является индуктивной нагрузкой, необходимо обеспечить защиту от пробоя. Если через обмотку течет ток, то при остановке мотора на выводах обмотки временно появляется всплеск напряжения, в силу 2-го закона коммутации. Это напряжение может привести к повреждению транзистора вызывая пробой перехода база-коллектор. Кроме того, это может создавать значительные помехи. Для предотвращения такого эффекта необходимо подключить диод параллельно с индуктивной нагрузкой, как на рисунке 5.

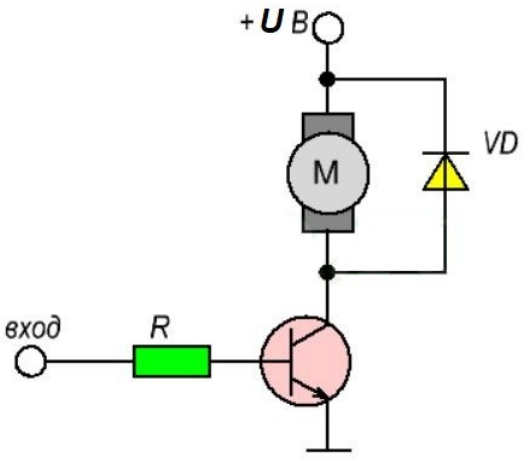


Рисунок 5

Во время нормальной работы двигателя диод смещен в обратном направлении. Отключение питания электродвигателя вызывает нарастание напряжения на катушке, при этом диод будет смещен в прямом направлении, благодаря чему произойдет разряд излишней энергии накопленной в катушке.

Диод следует подобрать такой, чтобы он выдерживал обратное напряжение во время нормальной работы двигателя. Такую защиту можно применять как при использовании биполярных транзисторов, так и MOSFET. Так же рекомендуется использовать диод и в работе с электромагнитным реле, для предотвращения раннего износа контактов.

Чтобы добавить диод в схему вовсе необязательно использовать дискретный диод. Вспомнив о том, что биполярный транзистор является прибором с двумя p-n переходами, то нетрудно догадаться, что такой транзистор можно включить в режиме диода, замкнув, например, выводы базы и эмиттера. Такая конфигурация является наиболее предпочтительной [12] в силу того, что база-коллекторный переход имеет в 10 более напряжение пробоя (50-60 В), чем база-эмиттерный (5-6 В). В таком случае анодом диода будет являться замкнутый контакт база-эмиттер, а катодом – коллектор.

Рекомендуемые источники:

1. Умняшкин С.В. Основы теории цифровой обработки сигналов: учебное пособие. Москва: ТЕХНОСФЕРА, 2019. - 550 с.
2. Сайт Espressif, информационный лист об API-функции АЦП для СнК ESP32 URL: https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html (дата обращения 20.03.2023)

3. Сайт Chipdip, информационный лист на прибор LM235Z URL: https://static.chipdip.ru/lib/908/DOC018908677.pdf (дата обращения 20.03.2023)
4. Сайт Espressif с документацией на TCP/IP: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/lwip.html (дата обращения 21.03.2023)
5. Сайт Espressif, информационный лист об API-функции ШИМ для управления LED для СнК ESP32 URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/ledc.html (дата обращения 20.03.2023)
6. Сайт Espressif, информационный лист об API-функции ШИМ для управления электродвигателем для СнК ESP32 URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/mcpwm.html (дата обращения 20.03.2023)
7. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ. – Изд. 2-е. – М.: Издательство БИНОМ. – 2016. – 704 с.
8. Сайт HABR «ШИМ в ESP32» URL: https://habr.com/ru/company/first/blog/664922/ (дата обращения 20.03.2023)
9. Сайт Электросам URL: https://electrosam.ru/glavnaja/jelektrotehnika/shirotno-impulsnaia-moduliatsiia/ (дата обращения 20.03.2023)
10. Сайт WikiAmperka URL: http://wiki.amperka.ru/articles:servo (дата обращения 21.03.2023)
11. Информационный лист для сервопривода TowerPro MG995 URL: https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf (дата обращения 21.03.2023)
12. Миндеева А.А. Микросхемотехника: учеб. пособие. - Изд. 2-е. - М.: МИЭТ, 2016. -188 с.