

リストA Cargo.tomlに依存を追加する

```
embedded-hal = "=1.0.0-alpha.9"
esp-idf-hal = "0.40.1"
bme280 = { git = 'https://github.com/quentinmit/bme280-rs.git', branch = 'async' }
```

リストB DelayUsトレイトのesp-idf-halでの実装

```
/// FreeRTOS-based delay provider
pub struct FreeRtos;

impl FreeRtos {
    fn delay_us_internal(&mut self, us: u32) {
        let ms = us / 1000;

        Self::delay_ms_internal(self, ms);
    }

    fn delay_ms_internal(&mut self, ms: u32) {
        // divide by tick length, rounding up
        let ticks = ms.saturating_add(portTICK_PERIOD_MS - 1) / portTICK_PERIOD_MS;

        unsafe {
            vTaskDelay(ticks);
        }
    }
}
```

図A bme280の実行結果

```
I (336) bme280: temperature: 21.501692, humidity: 90.29879, pressure: 66456.25
I (1376) bme280: temperature: 19.051517, humidity: 49.94993, pressure: 101528.414
I (2416) bme280: temperature: 19.051838, humidity: 50.165596, pressure: 101528.14
I (3456) bme280: temperature: 19.051838, humidity: 50.235535, pressure: 101527.97
I (4496) bme280: temperature: 19.051838, humidity: 50.0024, pressure: 101527.97
```

リストC Cargo.tomlに依存関係を追加する

```
[dependencies]
ssd1331 = "0.3"
embedded-graphics = "0.7"
```

リストD SPIドライバの初期化

```
let sdi_not_used: Option<Gpio0> = None;
let cs_not_used: Option<Gpio0> = None;

let spi = spi::SpiDeviceDriver::new_single(
    spi,
    sclk,
    sdo,
    sdi_not_used,
    spi::Dma::Disabled,
    cs_not_used,
    &config,
)?;
```

リストE no_std_anyhow!() マクロ

```
macro_rules! no_std_anyhow {
    ($e:expr) => {
        ($e).map_err(|e| anyhow!("{:?}, {:?}", e, std::file!(), std::line!()))
    };
}

fn main() -> anyhow::Result<()> {
    // 中略
    no_std_anyhow!(disp.reset(&mut rst, &mut delay))?;
}
```

リストF forループの中で1秒毎にセンサの値を読んでいる

```
for _ in 0..5 {
    let m = bme280
    .measure(&mut delay)
    .map_err(|e| anyhow!("{:?}, {:?}", e, std::file!(), std::line!()))?;
    info!(
        "temperature: {}, humidity: {}, pressure: {}",
        m.temperature, m.humidity, m.pressure
    );

    std::thread::sleep(std::time::Duration::from_secs(1));
}
```

図B BME280でセンサから値を読み出す

```
I (1338) timer: temperature: 21.501692, humidity: 90.29879, pressure: 66456.25
I (2338) timer: temperature: 18.89983, humidity: 56.791843, pressure: 101801.26
I (3338) timer: temperature: 18.89951, humidity: 56.791832, pressure: 101801.02
I (4338) timer: temperature: 18.89983, humidity: 56.756954, pressure: 101801.086
I (5338) timer: temperature: 18.900473, humidity: 56.722088, pressure: 101801.016
I (6338) timer: temperature: 18.900473, humidity: 56.70464, pressure: 101801.016
I (7338) timer: temperature: 18.901115, humidity: 57.803513, pressure: 101800.945
I (8338) timer: temperature: 18.901436, humidity: 57.9256, pressure: 101801
I (9338) timer: temperature: 18.901758, humidity: 57.92562, pressure: 101800.91
```

図C タイマが停止せず、動き続けるログ

```
I (8338) timer: temperature: 19.144714, humidity: 56.672863, pressure: 101795.375
I (9338) timer: temperature: 19.144714, humidity: 56.7485, pressure: 101795.2
I (10338) timer: temperature: 19.144394, humidity: 56.969566, pressure: 101795.18
I (11338) timer: temperature: 19.144394, humidity: 57.208084, pressure: 101795.18
I (12338) timer: temperature: 19.144394, humidity: 57.56874, pressure: 101795.18
```

リストG unsafeが必要な場合

```
// esp-idf-hal/src/gpio.rs
/// # Safety
///
/// Care should be taken not to call STD, libc or FreeRTOS APIs (except for a few al"L"レベルled ones)
/// in the callback passed to this function, as it is executed in an ISR context.
#[cfg(all(not(feature = "riscv-ulp-hal"), feature = "alloc"))]
pub unsafe fn subscribe(&mut self, callback: impl FnMut() + 'static) -> Result<(), EspError>
where
    MODE: InputMode,
{
    // 略
}
```

リストH esp-idf-hal/src/interrupt.rs抜粋

```

pub unsafe fn notify(task: TaskHandle_t, notification: u32) -> bool {
    let notified = if super::active() {
        let mut higher_prio_task_woken: BaseType_t = Default::default();

        let notified = xTaskGenericNotifyFromISR(
            task,
            0,
            notification,
            eNotifyAction_eSetBits,
            ptr::null_mut(),
            &mut higher_prio_task_woken as *mut _,
        );

        if higher_prio_task_woken != 0 {
            do_yield();
        }

        notified
    } else {
        let notified = xTaskGenericNotify(
            task,
            0,
            notification,
            eNotifyAction_eSetBits,
            ptr::null_mut(),
        );

        notified
    };

    notified != 0
}

```

図D 割り込みハンドラ内でinfo!()やprintln!()を呼び出したときのログ

```

I (279) gpio: GPIO[9]| InputEn: 0| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:0
# ここでボタンを押している
ESP-ROM:esp32c3-ap1-20210207
Build:Feb 7 2021
rst:0x3 (RTC_SW_SYS_RST),boot:0xc (SPI_FAST_FLASH_BOOT)
Saved PC:0x403812a6
0x403812a6 - esp_restart_noos_dig
    at /home/tomoyuki/.espressif/esp-idf/v4.4.1/components/esp_system/esp_system.c:44
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x16b4
load:0x403ce000,len:0x930
0x403ce000 - _iram_end
    at ????:??
load:0x403d0000,len:0x2d28
0x403d0000 - _iram_end
    at ????:??
SHA-256 comparison failed:
Calculated: f78487f03d5d24753454e5c7a8655bf4fb0c2a73f119fa8ece9b74b440a2f898
Expected: 54ea11eec255afa0b136329ec3d31790eee6510b85edfdc31b7ba8608f107751
Attempting to boot anyway...
entry 0x403ce000
0x403ce000 - _iram_end
    at ????:??

```

再起動している

I (42) boot: ESP-IDF v4.4.1 2nd stage bootloader

I (42) boot: compile time 05:32:19

リストI ESP-IDFでのMutex実装

```
static int IRAM_ATTR pthread_mutex_lock_internal(esp_thread_mutex_t *mux, TickType_t tmo)
{
    if (!mux) {
        return EINVAL;
    }

    if ((mux->type == PTHREAD_MUTEX_ERRORCHECK) &&
        (xSemaphoreGetMutexHolder(mux->sem) == xTaskGetCurrentTaskHandle())) {
        return EDEADLK;
    }

    if (mux->type == PTHREAD_MUTEX_RECURSIVE) {
        if (xSemaphoreTakeRecursive(mux->sem, tmo) != pdTRUE) {
            return EBUSY;
        }
    } else {
        if (xSemaphoreTake(mux->sem, tmo) != pdTRUE) {
            return EBUSY;
        }
    }

    return 0;
}
```