

0.5 – Program blink arduino dengan bahasa ada

Ketika saya sedang berselancar di internet, saya menemukan sebuah informasi bahwa ada bisa digunakan untuk embeded programming, seperti [0] post saya sebelumnya untuk bahasa c, kenapa tidak membuatnya untuk bahasa ada.

beberapa hari saya menggunakan ada, dan saya rasa ada sangat bagus, type system yang sangat kuat dan ada merupakan bahasa yang safe seperti rust, tetapi tanpa syntak yang aneh, ada termasuk kedalam keluarga [1] pascal jadi syntaknya termasuk mudah dibaca, saya menyukainya.

seperti cargo rust, sekarang ada mempunyai package manager yang bernama [2] alire, karna program alire ini kita dapat dengan mudah menggunakan library dan toolchain yang berbeda, untuk ini saya menggunakan toolchain [3] gnat_avr_elf.

setelah toolchain agar kita bisa membuat executable, saya memerlukan library, un- uk avr, terdapat 3 library yang kita perlukan yaitu [4] avrada_rts, [5] avrada_mcu dan [6] avrada_lib, sesuai namanya rts berisi run time system, mcu berisi data mcu untuk model apa yang saya pakai dan lib berisi library utilitas.

untuk compile ke avr kita perlu menseting model mcu sebagai target, dan juga clock speed target, setelah di setting seperti inilah file alire.toml

```
[configuration.values]
avrada_rts.AVR_MCU = "atmega32u4"
avrada_rts.Clock_Frequency = 16000000

[[depends-on]]
gnat_avr_elf = "^11 | ^12.2"
avrada_rts = "^2.0.1"
avrada_mcu = "^2.2.0"
avrada_lib = "^2.1.0"
```

setelah konfigurasi alire, kita bisa menggunakan perintah alr update agar semua library terinstall.

selanjutnya kita akan membuat sebuah gprbuild file, dan membuat sebuah program sederhana seperti LED blink, seperti inilah gprbuild file.

```
with "avrada_rts.gpr";
with "avr_tool_options.gpr";
with "avrada_mcu.gpr";
with "avrada_lib.gpr";

project Hello_Avr is

  for Target use "avr";
  for Runtime ("ADA") use AVRada_RTS'Project_Dir;

  for Source_Dirs use (".");
  for Object_Dir use "obj";
  for Create_Missing_Dirs use "True";

  for Exec_Dir use ".";
  for Main use ("hello_avr");

  package Builder renames AVR_Tool_Options.Builder;

  package Compiler is
    for Default_Switches ("Ada") use AVR_Tool_Options.ALL_ADAFLAGS;
  end Compiler;

  package Binder is
    for Switches ("Ada") use AVR_Tool_Options.Binder_Switches;
  end Binder;

  package Linker is
    for Switches ("Ada") use AVR_Tool_Options.Linker_Switches;
  end Linker;

end Hello_Avr;
```

menurut saya gpr mempermudah kita, kita hanya perlu membuat gpr file lalu mengetik `avr build`, maka executable akan langsung di buat, kita juga membuat agar package ADA dapat digantikan dengan AVR ADA.

setelah membuat gpr build kita membuat program utama kita, program BLINK LED untuk avr.

```
with AVR;
with AVR.MCU;
with AVR.Wait;
with Avrada_Rts_Config;

procedure Hello_Avr is
  LED : Boolean renames AVR.MCU.PORTC_Bits (7);

  procedure Wait_1_Sec is new
    AVR.Wait.Generic_Wait_USecs (Avrada_Rts_Config.Clock_Frequency,
                                  1_000_000);

begin
  loop
    LED := AVR.High;
    Wait_1_Sec;
    LED := AVR.Low;
    Wait_1_Sec;
  end loop;
end Hello_Avr;
```

seperti inilah kode yang kita gunakan untuk BLINK LED, dilihat-lihat program ada lebih mudah dibaca dibanding dengan bahasa c, di bahasa C kita menggunakan makro atau operator shift, contohnya di c saya menggunakan "DDRC |= _BV(DDC7);" tapi di ada kita menggunakan "AVR.MCU.PORTC_Bits (7)", setelah itu kita build menggunakan alr build, setelah itu akan ada .elf file, kita perlu mengubahnya dulu ke hex file dengan objcopy dan avrdude untuk upload seperti yang kita lakukan di post bahasa C saya

karna readable inilah, mungkin semua projek saya akan menggunakan bahasa ada dibanding bahasa c.

Terima kasih atas waktu yang telah Anda habiskan di wiki ini, tetaplah terhubung untuk pembaruan lebih lanjut, Sampai Jumpa.