Hướng dẫn nạp code cho máy in 3D và mở khoá bootloader cho bo mạch MKS Robin V2.4 + màn hình TFT V2.0

Hướng dẫn thuộc Facebook Mr Anthony Trần, Printer QH – Hà Nội – 0962863070

Bài viết có thể viết không tránh khỏi sai sót và câu chữ có thể còn khó hiểu xin mọi người đóng góp thêm ý kiến vì mình viết theo tư duy cá nhân.

1 – Tại sao phải mở khoá bootloader dòng chip và bo mạch này ?

Hệ chip trên bo mạch mình đang Bootloader là STM32F103ZET6 đây là hệ thống bo mạch đã được sản xuất cách đây khá lâu và nó vẫn còn khá là mạnh mẽ và hoạt động ổn định, có thêm chức năng Wifi cho việc dùng APP điều khiển qua địa chỉ IP. Hoặc thiết lập một hệ sinh thái cho cả một dàn máy in 3D

- Khoá Bootloader là gì : Khoá Bootloader là trạng thái các vùng làm việc và các thành phần trên bo mạch bị hạn chế can thiệp vào hệ thống, và bạn không thể tuỳ ý tuỳ biến hay sửa chữa các chức năng, hay thêm các chức năng khác vào bo mạch. Bạn bắt buộc bị lệ thuộc vào cấu hình và code của nhà sản xuất cung cấp cho bạn.
- Mở khoá Bootloader là gì : Tương tự như việc Lock Boot thì Unlock Bootloader giúp ta có thể can thiệp sâu vào hệ thống tuỳ chỉnh thoải mái các chức năng code, thêm hay xoá các chương trình và cấu hình các chức năng mở rộng hay thêm bất kỳ một đoạn code nào mà chúng ta cần thêm vào chương trình.

2 – Có nhiều bạn hỏi mình là tại sao phải làm như này sao phải rườm rà phức tạp hoá vấn đề.

- Thứ nhất mình là người thích tìm hiểu và tối ưu hoá các chức năng và chương trình trong code mà nhà sản xuất còn hạn chế. Mình thích tự Build và tự chế tạo ra cái sản phẩm của riêng mình. Mình chỉ lệ thuộc vào nền tảng code là Open và từ đó phát triển ra cái mình cần và phát triển thêm những chức năng mình muốn có.
- Thứ hai đó là mình muốn các bạn hiểu thêm về các dạng bo mạch trên thị trường máy in 3D, biết phân biệt được bo mạch nào tốt và bo mạch nào kém chất lượng.
- Thứ ba mình muốn các bạn có thêm kiến thức và nền tảng khi tối ưu hoá code của mình khi các bạn tự làm. Hay khi các bạn muốn MOD thêm chức năng cho nó hay tự đóng Chip thay thế nó khi nó bị lỗi
- Thứ bốn giúp các bạn có thể hình dung được vấn đề lập trình code hay viết code nó như nào
- Thứ năm đây là bo mạch có thể nói đó là đứa con đầu đời của MKS BASE mình không biết vì lý do gì mà nhà sản xuất lại Lock chip này, còn các thế hệ bo mạch về sau phát triển từ nền tảng STM32 khác và LPC 176x thì họ không còn Lock chip nữa. Các bạn có thể tuỳ ý sử dụng nền tảng code Open để Build code theo ý muốn.

3 – Mình không chia sẻ mã nguồn của chương trình Unlock Bootloader và Frimware mà mình đang sử dụng mà mình chỉ hướng dẫn cách thức các bạn làm nó như nào.

Tất cả các file unlock hay code mình đều đã mã hoá nó lên các bạn chỉ việc copy nó vào thẻ nhớ để nạp code là có thể sử dụng được. Bộ file được chia sẻ trong Group của mình

https://www.facebook.com/groups/492944117736285

Kênh Youtube của mình

https://www.youtube.com/channel/UCtytbQ4-hvg0s1FLdeq3kSg

4 – Cách thức làm

- Bo mạch MKS robin V2.4 + Mks robin TFT 2.0
- Thẻ nhớ Format về định dạng FAT32
- Cáp USB
- Nguồn điện 12V 30A, hệ thống Driver A4988 hoặc TMC tuỳ các bạn dùng, động cơ bước, công tắc hành trình, dây nung nhiệt, cảm biến nhiệt, bàn nhiệt để test máy khi Build code song

5 – Phần mềm

- Visula Studio Code được cài Platform IO và máy tính được cài Python phiên bản mới nhất
- Bộ code Open Marlin *(Dùng bản bug fix mới nhất)*
- Hoặc có thể dùng phần mềm Atom hoặc Sublime Text 3 để biên dịch code. Tuỳ cách thức mỗi người mà bạn dùng phần mềm nào thì bạn có thể dùng nó
- Cách dùng chương trình biên dịch này các bạn lên Youtube nó có rất nhiều hướng dẫn mình không hướng dẫn

6 – Thực hiện chương trình

- Copy file Robin.bin có dung lượng là 21Kb vào thẻ nhớ sau đó lắp thẻ nhớ vào bo mạch rồi bật điện nguồn lên thấy màn hình TFT báo đang Update sau khi nó Update song thì thấy đèn LED màu xanh nó nhấp nháy gần chỗ cắm Module Wifi là được. Lúc này ta đã Unlock Bootloader cho bo mạch song.
- Dowload bộ code Marlin về giải nén nó ra một ổ đĩa hay một thư mục nào đó, sau đó dùng chương trình Visula hay Atom và Sublime Text 3 đã được nhưng đã được cài Platform IO và dùng Platform để mở code
- Sau đó tìm chương trình platformio. ini thay đổi dòng default_envs thành mks_robin
- Tiếp tục vào chương trình cấu hình cho bo mạch của Marlin thay đổi các dòng sau thành đúng giá trị như này.
- Cấu hình chương trình cho máy in 3D mình không hướng dẫn vì có quá nhiều hướng dẫn rồi các bạn tìm trên Google.
-

```
-   #define SERIAL_PORT 3
-
-   #define BAUDRATE 250000
-   #define BAUD_RATE_GCODE
-
-   #define SERIAL_PORT_2 1
-   #define NUM_SERIAL 2
-
-   #ifndef MOTHERBOARD
-     #define MOTHERBOARD BOARD_MKS_ROBIN
-   #endif
-
-   #define SDSUPPORT
```

```
-   #define SDIO_SUPPORT
-
-   // 320x240, 3.2", FSMC Display From MKS
-   // Normally used in MKS Robin Nano V1.2
-   //
-   #define MKS_ROBIN_TFT32 // Lựa chọn TFT này cho MKS Robin V2.4 để hiển thị V2.0 độ phân
    giải 320x240 cho V2.0
-
-   #define TFT_COLOR_UI
-
-   Sau đó sao chép nguyên đoạn code này dán vào phần TFT hoặc kéo xuống đoạn cuối cùng của
    code rồi Paster nó vào, đây là phần code mình thêm cho màn TFT bắt buộc phải thêm. Trong
    phần này mình viết bằng tiếng việt có dấu để cho các bạn dễ hiểu nội dung từng dòng code
    mình thêm vào, khi biên dịch code nó sẽ biên dịch thấy các ký tự loằng ngoằng các bạn có
    thể xoá hoặc viết lại tiếng việt không dấu hoặc tiếng anh thì tuỳ chọn.
-
-   //=============================================================================
    ============================================================================
-   //============================== Thiết lập giao diện màn hình Cảm ứng TFT Cho MKS Robin
    V2.4 với TFT 2.0, bản FW thuộc Mr Anthony Trần   ============================
-   //=============================================================================
    ============================================================================
-   // Trong quá trình biên dịch code nếu gặp các lỗi hiển thị ký tự lạ thì do mình đang viết
    tiếng việt có dấu này giúp các bạn dễ hiểu vấn đề
-   // Lựa Chọn TFT cho Mks Robin V2.4
-   // Phải mở tính năng Color Marlin UI cho Mks Robin và TFT
-   //
-   //#define TFT_320x240
-   //#define TFT_320x240_SPI
-   //#define TFT_480x320
-   //#define TFT_480x320_SPI
-
-   //
-   // Chế độ màn cảm ứng phải mở toàn bộ dòng này ra
-   //
-   #define TOUCH_SCREEN
-   #if ENABLED(TOUCH_SCREEN)
-     #define BUTTON_DELAY_EDIT  75 // (ms) Độ trễ khi di chuyển các lệnh ví dụ như tăng giảm
    nhiệt độ hoặc tăng giảm các giá trị Proble Offset, di chuyển số bước tiến mm/ step
-     #define BUTTON_DELAY_MENU 100 // (ms) Độ trễ khi di chuyển các chức năng của MENU trong
    TFT
-
-     #define TOUCH_SCREEN_CALIBRATION
-
-     /* Nếu bạn dùng MKS Robin TFT v2.0 thì mở hết dòng này ra, các giá trị căn chỉnh này
    không sửa chỉ có mình có mới chỉnh sửa được */
-     #define XPT2046_X_CALIBRATION    12013
-     #define XPT2046_Y_CALIBRATION    -8711
-     #define XPT2046_X_OFFSET          -32
-     #define XPT2046_Y_OFFSET          256
-
-     /* Nếu bạn dùng  MKS Robin TFT v1.1 thì mở hết dòng này ra, các giá trị căn chỉnh này
    không sửa chỉ có mình có mới chỉnh sửa được  */
-     //#define XPT2046_X_CALIBRATION -11792
-     //#define XPT2046_Y_CALIBRATION   8947
-     //#define XPT2046_X_OFFSET         342
```

```
    //#define XPT2046_Y_OFFSET         -19

    /* Nếu bạn dùng MKS Robin TFT v1.1 thì mở hết dòng này ra, các giá trị căn chỉnh này không sửa chỉ có mình có mới chỉnh sửa được  */
    //#define XPT2046_X_CALIBRATION  12489
    //#define XPT2046_Y_CALIBRATION   9210
    //#define XPT2046_X_OFFSET         -52
    //#define XPT2046_Y_OFFSET         -17
  #endif

  Sau khi thực hiện đầy đủ các yêu cầu ở trên thì các bạn cấu hình code theo ý muốn của các bạn như Driver, bước tiến, tốc độ in, các chế độ khác rồi bấm biên dịch code. Khi biên dịch song nó sẽ có 1 file là Robin.bin nó nằm ở đường dẫn này

  .pio\build\mks_robin tệp này được tạo ra trong quá trình biên dịch code và nó nằm ở trong bộ code bạn đang sử dụng
  Sau đó sao chép File Robin.bin vào thẻ nhớ và cắm vào bo mạch và Flask code khi flash code song nó sẽ có giao diện hiển thị như bài đăng này của mình

  https://www.facebook.com/groups/492944117736285/permalink/1556275831403103/?notif_id=1633961344392201&notif_t=page_post_reaction&ref=notif
```

7 – Kiểm tra các tập tin này ở đường dẫn này nó phải khớp với đoạn code này của mình. Nếu không khi biên dịch code nó sẽ báo lỗi. Lý do vì mỗi một bản Update code marlin nó đều bị thay đổi

buildroot\share\PlatformIO\scripts

common-dependencies.py kiểm tra xem chỗ này có bị sai không nó phải khớp với đoạn code này của mình

```python
#print(env.Dump())

try:
    verbose = int(env.GetProjectOption('custom_verbose'))
except:
    verbose = 0

def blab(str,level=1):
    if verbose >= level:
        print("[deps] %s" % str)

FEATURE_CONFIG = {}

def add_to_feat_cnf(feature, flines):

    try:
        feat = FEATURE_CONFIG[feature]
    except:
        FEATURE_CONFIG[feature] = {}

    # Get a reference to the FEATURE_CONFIG under construction
    feat = FEATURE_CONFIG[feature]

    # Split up passed lines on commas or newlines and iterate
    # Add common options to the features config under construction
    # For lib_deps replace a previous instance of the same library
    atoms = re.sub(r',\\s*', '\n', flines).strip().split('\n')
    for line in atoms:
        parts = line.split('=')
        name = parts.pop(0)
        if name in ['build_flags', 'extra_scripts', 'src_filter', 'lib_ignore']:
            feat[name] = '='.join(parts)
            blab("[%s] %s=%s" % (feature, name, feat[name]), 3)
        else:
            for dep in re.split(r",\s*", line):
                lib_name = re.sub(r'@([~^]|[<>]=?)?[\d.]+', '', dep.strip()).split('=').pop(0)
                lib_re = re.compile('(?!^' + lib_name + '\\b)')
                feat['lib_deps'] = list(filter(lib_re.match, feat['lib_deps'])) + [dep]
                blab("[%s] lib_deps = %s" % (feature, dep), 3)

def load_config():
    blab("========== Gather [features] entries...")
    items = ProjectConfig().items('features')
    for key in items:
        feature = key[0].upper()
        if not feature in FEATURE_CONFIG:
            FEATURE_CONFIG[feature] = { 'lib_deps': [] }
        add_to_feat_cnf(feature, key[1])

    # Add options matching custom_marlin.MY_OPTION to the pile
    blab("========== Gather custom_marlin entries...")
    all_opts = env.GetProjectOptions()
    for n in all_opts:
        key = n[0]
        mat = re.match(r'custom_marlin\.(.+)', key)
        if mat:
```

```
            try:
                val = env.GetProjectOption(key)
            except:
                val = None
            if val:
                opt = mat.group(1).upper()
                blab("%s.custom_marlin.%s = '%s'" % ( env['PIOENV'], opt, val ))
                add_to_feat_cnf(opt, val)

def get_all_known_libs():
    known_libs = []
    for feature in FEATURE_CONFIG:
        feat = FEATURE_CONFIG[feature]
        if not 'lib_deps' in feat:
            continue
        for dep in feat['lib_deps']:
            known_libs.append(PackageSpec(dep).name)
    return known_libs

def get_all_env_libs():
    env_libs = []
    lib_deps = env.GetProjectOption('lib_deps')
    for dep in lib_deps:
        env_libs.append(PackageSpec(dep).name)
    return env_libs

def set_env_field(field, value):
    proj = env.GetProjectConfig()
    proj.set("env:" + env['PIOENV'], field, value)
```

Sau khi kiểm tra đầy đủ các tính năng và chương trình thì chúc AE đã trở thành một người hiểu về code máy in 3D và có thể tự tạo cho mình một bộ code theo ý thích rồi đó. Chúc AE thành công

Xin lỗi nếu có điều gì cần mình giải đáp hay giúp đỡ xin liên hệ Zalo 0962863070 hoặc gọi điện, thời gian mình khá hạn chế lên AE gửi hình ảnh và điều cần giúp đỡ xin nói rõ thông tin và yêu cầu đi vào mục đích chính mình sẽ bớt chút thời gian giúp các bạn.

Ngoài ra mình có nhận Build các dạng máy in 3D hoặc lập trình code máy in 3D, tiến tới mình ra một số sản phẩm mới về máy in 3D mong AE ủng hộ mình.

Nếu không muốn config bản code này của mình mà muốn hoàn nguyên quay về bản code của nhà sản xuất thì các bạn vẫn có thể dùng bản config.txt và file Bin của hãng để flash lại code nó sẽ trở về giao diện mặc định.