

CCS C Derleyicisinde #export ve #hexcomment Kullanılarak Firmware HEX Versiyonlama

Yazar: Ertan Suluagaç

Unvan: Gömülü Yazılım Geliştirme Uzmanı

Tarih: 13 Aralık 2025

İletişim: ertansuluagac@windowsslive.com

Abstract

Firmware geliştirme süreçlerinde HEX dosyalarının doğru şekilde isimlendirilmesi, sürüm takibi ve üretim doğruluğu açısından kritik bir gereksinimdir. Varsayılan derleme çıktılarının proje adına bağlı olarak tek bir isimde (örneğin main.hex) üretildiği senaryolarda, farklı test sürümleri ve güncellemeler birbirine kolayca karışabilmektedir. Bu çalışmada, CCS C derleyicisinde bulunan #export ve #hexcomment direktifleri kullanarak HEX çıktıların dinamik ve izlenebilir biçimde isimlendirilmesi gösterilmektedir. Önerilen yaklaşım; ürün/uygulama adı, major-minor sürüm bilgisi ve derleme tipine göre (örnek: test veya yayım) dosya adının otomatik oluşmasını sağlar. Yöntem, basit bir örnek proje üzerinden derlenmiş çıktılar ile doğrulanmış; dosya adı ve HEX içeriğine gömülen açıklayıcı yorumların sahada yanlış firmware yükleme riskini azaltabileceği gösterilmiştir.

Keywords

firmware; HEX file; versioning; CCS C compiler; #export; #hexcomment

1. Introduction

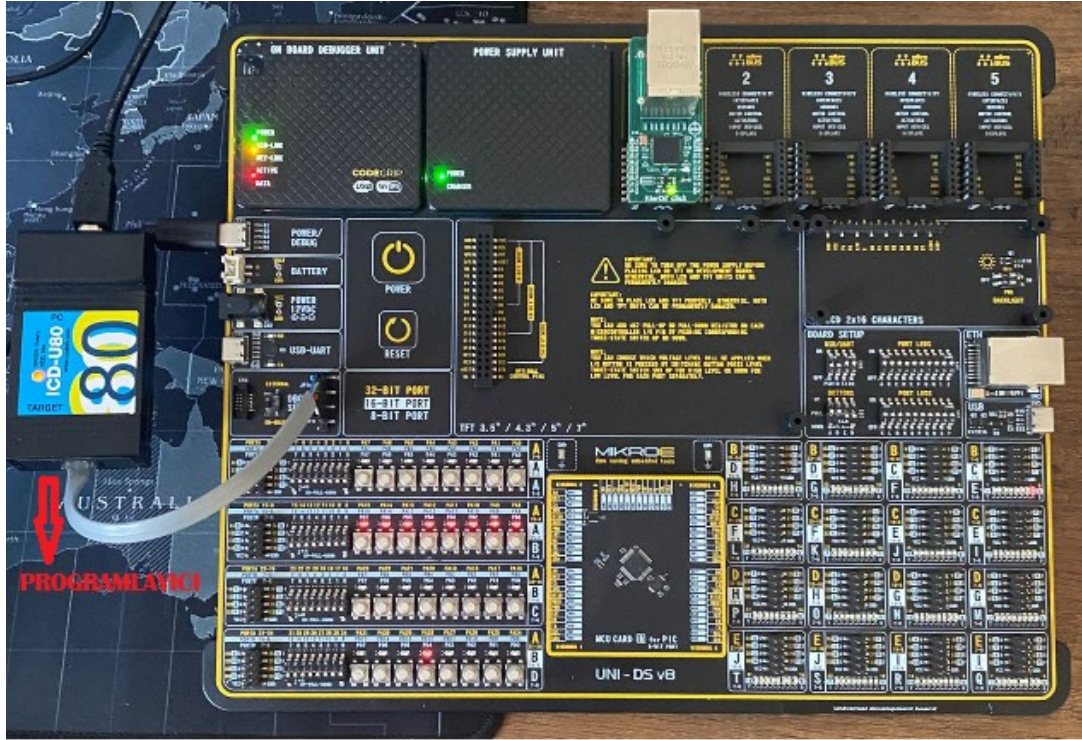
Gömülü sistem geliştirme süreçlerinde firmware dosyalarının izlenebilirliği, yalnızca kaynak kodun sürümlenmesi ile sınırlı değildir. Üretim hattı, servis ve saha güncellemeleri gibi pratik akışlarda, doğru firmware dosyasının hızlıca tespit edilmesi ve yanlış dosya seçiminin önlenmesi gerekir. Buna karşın, birçok projede derleme çıktıları varsayılan olarak proje adına bağlı tek bir isimde (örneğin main.hex) üretilir. Zamanla, klasörde biriken main (2).hex, final.hex gibi dosyalar hem takip zorluğu yaratır hem de insan hatasına açık bir ortam oluşturur.

Bu çalışmanın amacı, CCS C derleyicisinde yer alan #export direktifi ile HEX dosyası adının doğrudan kod içerisinden belirlenmesini ve #hexcomment ile dosyanın içine sürüm/uygulama bilgisi gömülmesini akademik bir düzende sunmaktır. Hedef;

okunabilir dosya adları (örneğin myproject_1.0.hex) üretmek, sürüm değiştiğinde otomatik olarak yeni bir çıktı oluşturmak ve sahada yanlış firmware yükleme riskini azaltmaktır.

2. Methodology

Bu bölümde, CCS C derleyicisinde HEX çıktı isimlendirmesini kontrol eden #export direktifi ve HEX dosyasına açıklayıcı yorum ekleyen #hexcomment direktifi açıklanmıştır. Yöntem, aşamalı olarak en basit kullanımdan başlayıp major-minor sürüm numarası ekleme ve makro değerlerini string'e çevirme tekniği ile genişletilmiştir.



Şekil 1 CCS ICD-U80 programlayıcı ve test ortamı

2.1. HEX dosyası adını #export ile belirlemek

En basit durumda, CCS derleyicisine oluşturulacak HEX dosyasının adını doğrudan bildirmek yeterlidir. Aşağıdaki örnek, derleme çıktı dosyasını myproject.hex olarak oluşturur.

```
#export(file="myproject.hex", hex)
```

```
#include <18F57Q43.h>
```

```

#device ADC=12

#FUSES NOWDT                //No Watch Dog Timer

#use delay(clock=64MHz,crystal=16MHz)

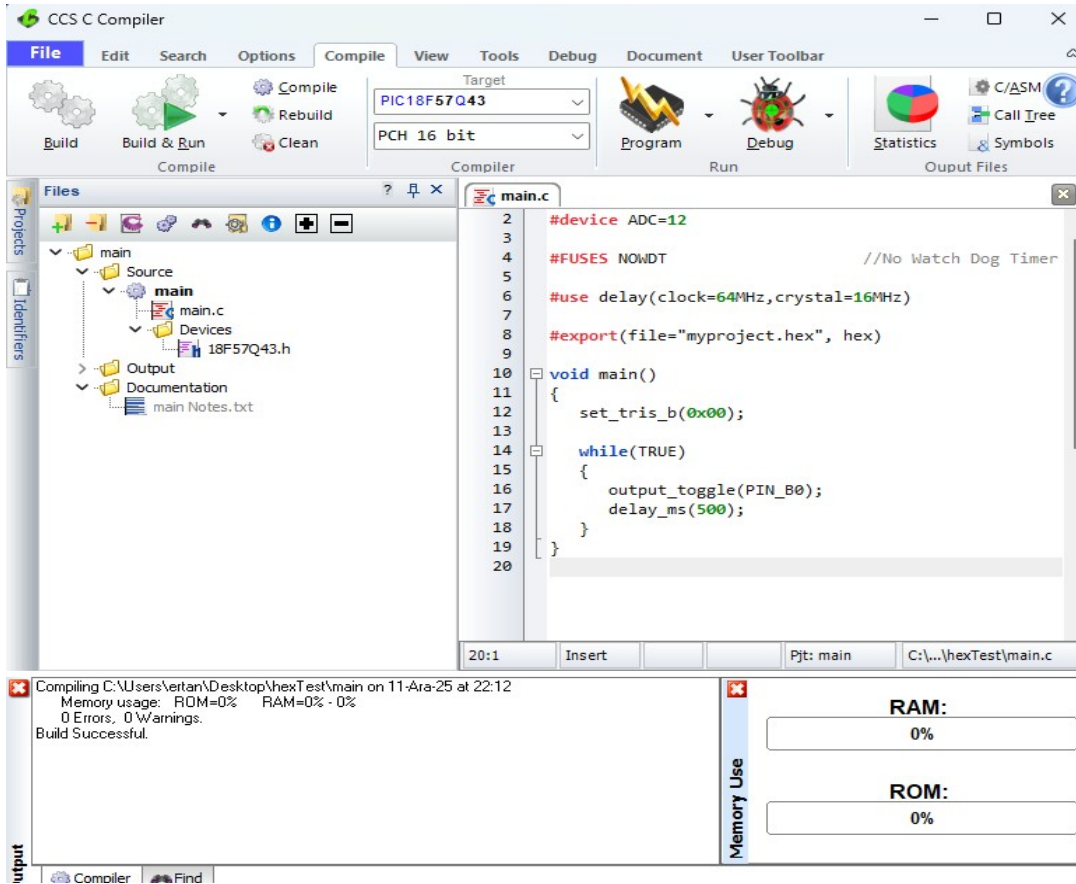
#export(file="myproject.hex", hex)

void main()
{
    set_tris_b(0x00);

    while(TRUE)
    {
        output_toggle(PIN_B0);
        delay_ms(500);
    }
}

```

Bu yaklaşımla, proje adı main olsa dahi derleme çıktı dosyasının adı değişmez; firmware dağıtımı ve arşivleme açısından tekil ve kontrol edilebilir bir çıktı elde edilir.



Şekil 2 Derleme sonrası myproject.hex dosyası

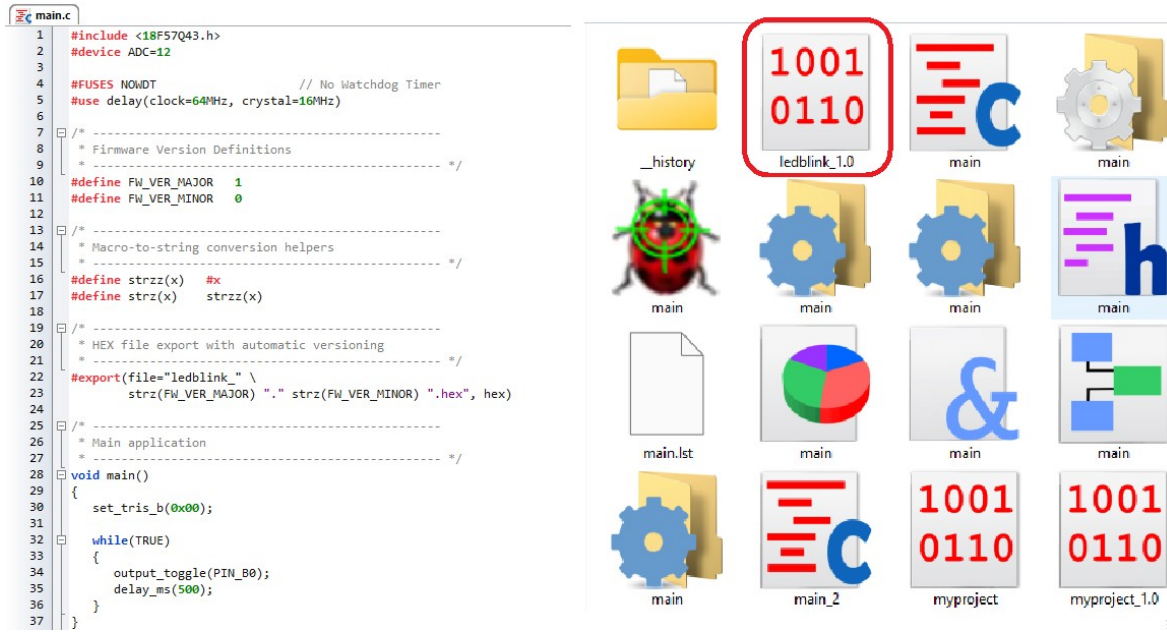
```
/* -----  
 * Firmware Version Definitions  
 * ----- */  
#define FW_VER_MAJOR    1  
#define FW_VER_MINOR    0  
  
/* -----  
 * Macro-to-string conversion helpers  
 * ----- */  
#define strzz(x)        #x  
#define strz(x)         strzz(x)
```

```

/* -----
 * HEX file export with automatic versioning
 * ----- */
#export(file="ledblink_" \
        strz(FW_VER_MAJOR) "." strz(FW_VER_MINOR) ".hex", hex)

```

Bu yapı ile derleyici, dosya adını derleme aşamasında oluşturur ve sonuç olarak ledblink_1.0.hex dosyası üretilir. Bu işlem çalışma zamanında ek maliyet oluşturmaz; tamamıyla derleme aşamasında gerçekleşir.



Şekil 4 Versiyonlu HEX dosyaları klasör görünümü

2.4. HEX dosyasına bilgi gömmek: #hexcomment

Dosya adı birçok durumda yeterli olsa da, HEX içeriğine ek bilgi gömmek üretim ve servis süreçlerinde ek bir güvenlik katmanı sağlar. `#hexcomment` direktifi, programlayıcılar tarafından yok sayılan bir yorum satırını HEX dosyasına ekler. Aşağıdaki örnek, dosya sonuna yorum ekler:

```

/* Hex file identification (written at end of HEX file) */
#hexcomment\ Created by: Ertan Suluagac (CCS C)

```

Not: `#hexcomment` sonundaki ters bölü (`\`) karakteri, yorumun dosya sonuna yazılmasını sağlar. Ters bölü kullanılmazsa yorum dosyanın başına eklenir.

```
ledblink_1.0.hex
Dosya Düzenle Görünüm

|:10000001AEF00F0050EEA6E060EE96EEF50D8B456
:1000100018EF00F0140E016E006A002EFED7012EBC
:10002000FBD7BF0E006E002EFED700D0EF2EF2D70A
:100030003BEF00F0F86AD69AB06BAF6BB36BB16B65
:10004000200EAD6F0401006B086B106B186B206BFA
:10005000286B0001716B726B736B706B756B766BD9
:10006000776B746B000EC76EC790BF70020E056E83
:0E007000FA0E066EC7D7052EFBD7F6D7030093
:020000040030CA
:0A000000AAFFF7DF9FFFFFFFDD
:00000001FF
; Created by: Ertan Suluagac (CCS C)
;PIC18F57Q43
;CRC=F2A1 CREATED="14-Ara-25 13:15"
#hexcomment\ Created by: Ertan Suluagac (CCS C)
```

Kullanım şekli / yukarıdaki gibidir

```
ledblink_1.0.hex
Dosya Düzenle Görünüm

; Created by: Ertan Suluagac (CCS C)
:100000001AEF00F0050EEA6E060EE96EEF50D8B456
:1000100018EF00F0140E016E006A002EFED7012EBC
:10002000FBD7BF0E006E002EFED700D0EF2EF2D70A
:100030003BEF00F0F86AD69AB06BAF6BB36BB16B65
:10004000200EAD6F0401006B086B106B186B206BFA
:10005000286B0001716B726B736B706B756B766BD9
:10006000776B746B000EC76EC790BF70020E056E83
:0E007000FA0E066EC7D7052EFBD7F6D7030093
:020000040030CA
:0A000000AAFFF7DF9FFFFFFFDD
:00000001FF
;PIC18F57Q43
;CRC=104F CREATED="14-Ara-25 13:20"
```

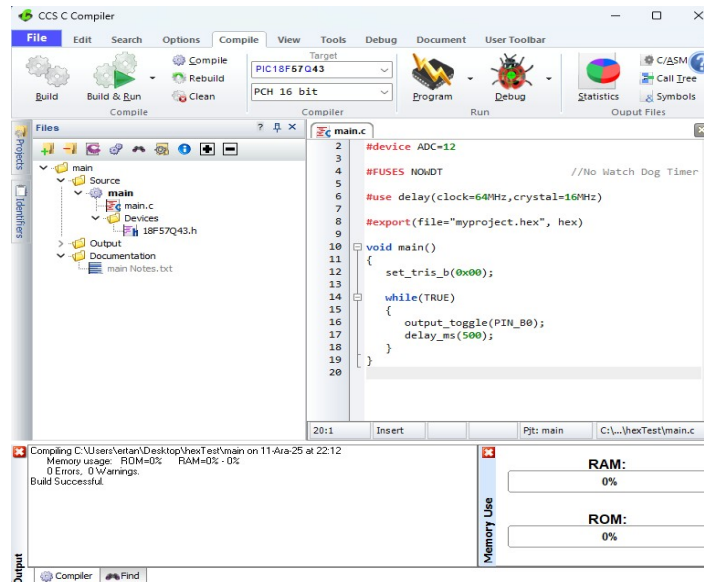
Kullanım şeklinde / yok

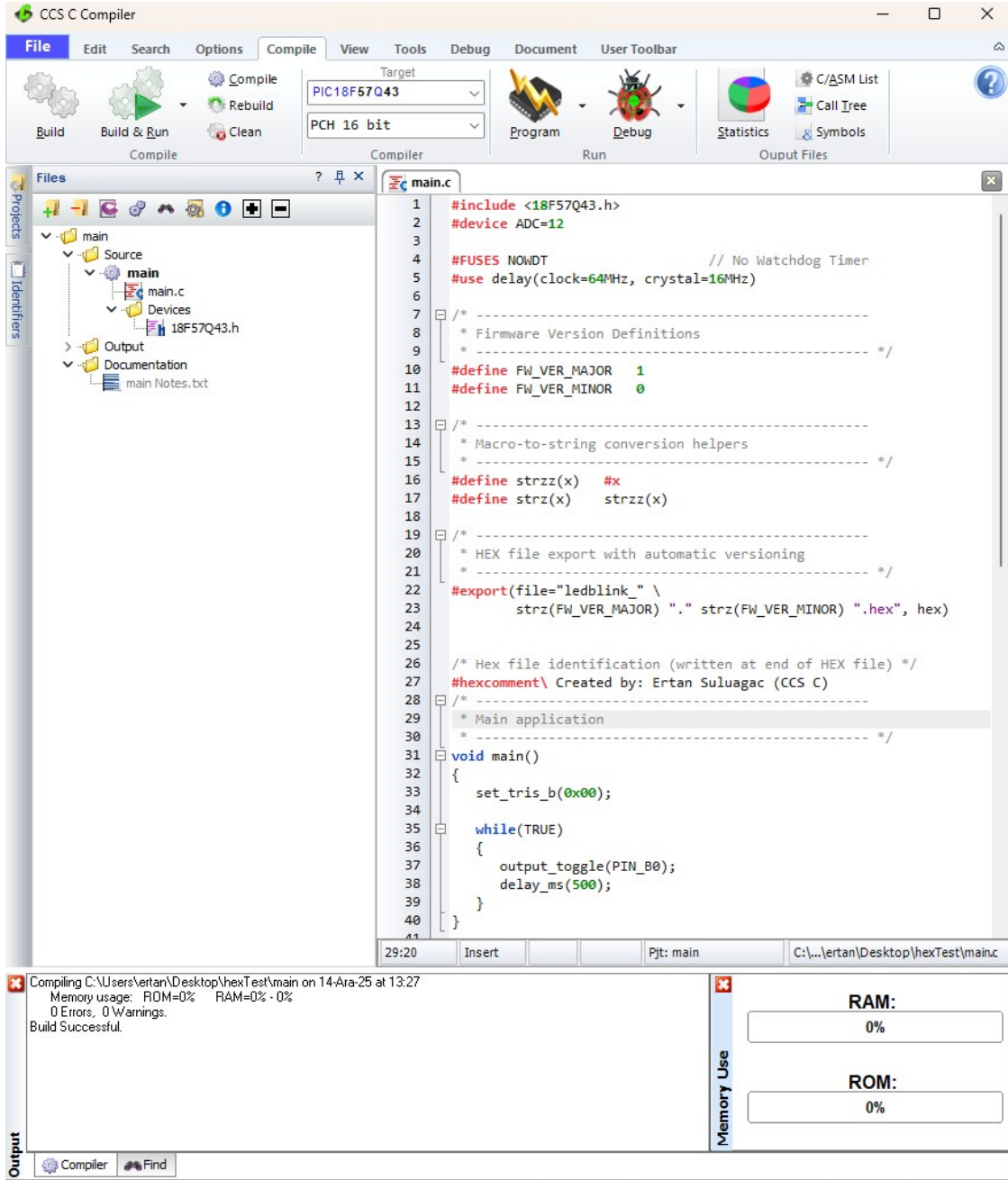
Şekil 5 #hexcomment direktifi kullanılarak HEX dosyasına gömülen açıklayıcı metadata örneği

3. Results

Yöntemin doğrulanması için basit bir LED blink projesi derlenmiş ve çıktı dosyaları incelenmiştir. Bu bölümde, derleme sonucu oluşan dosya adları ve HEX içeriği üzerinden elde edilen bulgular sunulmaktadır.

İlk derlemede #export ile sabit ad kullanılarak myproject.hex dosyası üretilmiştir. CCS IDE derleme çıktı penceresinde başarılı derleme mesajı görülmüş ve proje klasöründe myproject.hex dosyası oluşmuştur.

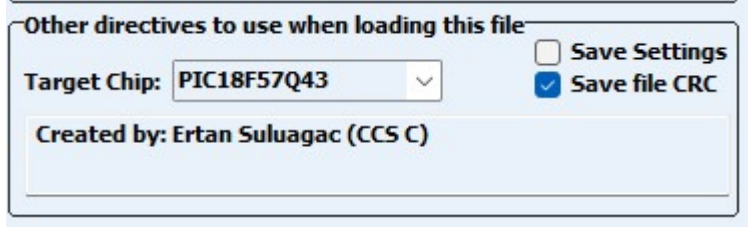




Şekil 6 CCS IDE derleme çıktısı (ledblink_1.0.hex oluşumu)

İkinci adımda, versiyon bilgisi dosya adına eklenmiş ve `myproject_1.0.hex` dosyası elde edilmiştir. `FW_VER_MINOR` değeri 1 yapıldığında, yeni derlemede `ledblink_1.0.hex` üretilmiş; bu sayede sürümler dosya adından doğrudan ayırt edilebilir hale gelmiştir.

Ek olarak `#hexcomment` ile HEX dosyasına eklenen yorum satırı, programlama işlemini etkilemeden dosya içinde sürüm bilgisinin taşınmasını sağlamıştır. HEX dosyası metin editörü ile açıldığında yorum satırı dosya sonunda görülebilmektedir.



Şekil 7 HEX dosyasında #hexcomment ile eklenen metadata satırının görünümü.

Not: **İlerleyen deneylerde**, otomatik sürüm numaralı isimlendirme yaklaşımının etkisini daha net gösterebilmek amacıyla, örnek proje adı **“ledblink”** olarak değiştirilmiştir.

4. Discussion and Limitations

Bu çalışmada sunulan yaklaşım, firmware dağıtımında en sık görülen problem olan dosya karışıklıklarını düşürmeyi hedefler. Dosya adının kod içerisinden kontrol edilmesi, "proje adı = HEX adı" bağımlılığını kaldırarak üretim ve servis süreçlerinde daha net bir izlenebilirlik sağlar. Versiyonun dosya adına eklenmesi, sahadaki teknisyen veya mühendis için doğru dosyayı seçmeyi kolaylaştırır.

Bununla birlikte yaklaşımın sınırlı yönleri vardır. Birincisi, burada gösterilen direktifler CCS C derleyicisine özgüdür; farklı toolchain'lere taşımak için eşdeğer mekanizmalar araştırılmalıdır. İkinci olarak, yalnızca dosya adına bakarak tüm değişikliklerin izlenmesi yeterli olmayabilir; kritik uygulamalarda kaynak kod sürüm etiketleri (tag), derleme tarihi ve donanım revizyonu gibi ek bilgilerle birlikte kullanılması önerilir. Bu noktada #hexcomment kullanımı, dosya adına ek olarak bir kimlik katmanı sağlar; ancak kurum içi bir isimlendirme standardının yazılı hale getirilmesi yine de gereklidir.

5. Conclusions and Future Work

Bu çalışma, CCS C derleyicisinde #export ve #hexcomment direktifleri kullanarak HEX dosyalarının akıllı ve izlenebilir biçimde üretilebileceğini göstermiştir. Basit bir örnek proje üzerinden, myproject.hex şeklinde sabit isimlendirme; ardından myproject.hex / myproject_1.0 / ledblink_1.0.hex şeklinde versiyonlu isimlendirme başarılı biçimde uygulanmıştır. HEX dosyası içerisine metadata gömülmesi, saha ve üretim senaryolarında yanlış firmware seçimini azaltabilecek pratik bir yaklaşımdır.

Gelecek çalışmalar kapsamında; otomatik derleme sistemleri (CI), dosya adına tarih veya commit etiketi eklenmesi, donanım revizyonuna göre isimlendirme ve

retim hattında ek doęrulama adımları (rneęin imza veya checksum) gibi geliřmiř nlemler ele alınabilir. Bu iyileřtirmeler, firmware daęıtım srecini daha da gvenli ve izlenebilir hale getirecektir.

References

1. CCS, Inc. (2021). Hex File Names. CCS News, 14 October 2021.
2. CCS C Compiler Manual / User Guide (ilgili blmler).