## Özet

Bu rapor, MSP430 G2553 mikrodenetleyici için geliştirilmiş olan assembler toolchain'inin yeni modülleri olan makro sistemi, ELF linker ve flash loader'ın teknik özelliklerini ve implementasyon detaylarını sunmaktadır. Proje kapsamında, mevcut iki geçişli assembler altyapısına entegre edilen bu yeni modüller, modern assembler toolchain'lerinin standart özelliklerini sağlamakta ve MSP430 G2553'ün özel gereksinimlerini karşılamaktadır.

## 1. Giriş

MSP430 G2553 için daha önce geliştirilen iki geçişli assembler sistemine, endüstriyel standartlarda bir toolchain oluşturmak amacıyla üç yeni kritik modül eklenmiştir:

1. **Makro İşlemci (macro.py)**: TI MSP430 standart makro formatını destekleyen gelişmiş makro sistemi
2. **ELF Linker (msp430\_linker.py)**: Çoklu object dosyasını birleştiren ve relocatable ELF formatını destekleyen linker
3. **Flash Loader (msp430\_loader.py)**: ELF executable'ları MSP430 flash memory'sine yükleyen programlama aracı

Ayrıca, mevcut assembler'da ELF object formatı desteği (elf\_object\_writer.py) ve GUI geliştirmeleri yapılmıştır.

## 2. Makro Sistemi (macro.py)

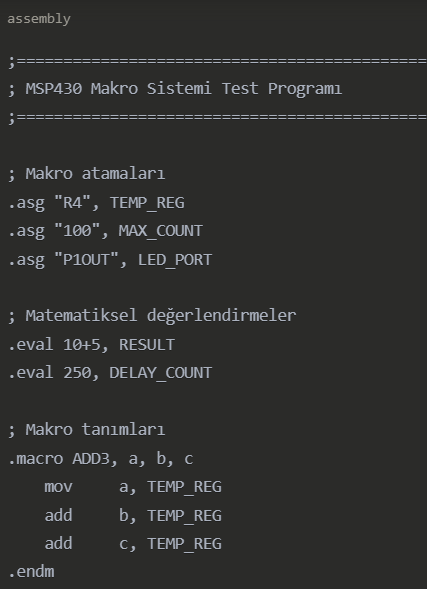
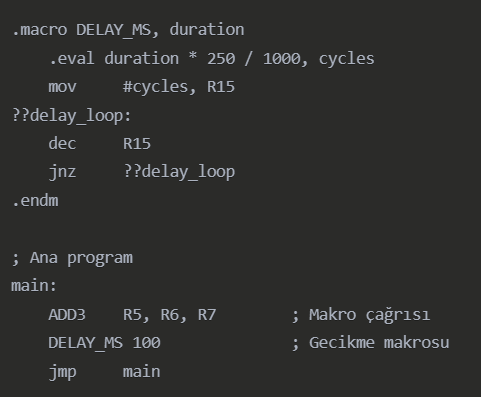
### 2.1 Teknik Özellikler

Makro sistemi, TI MSP430 standart makro formatını tam olarak desteklemektedir:

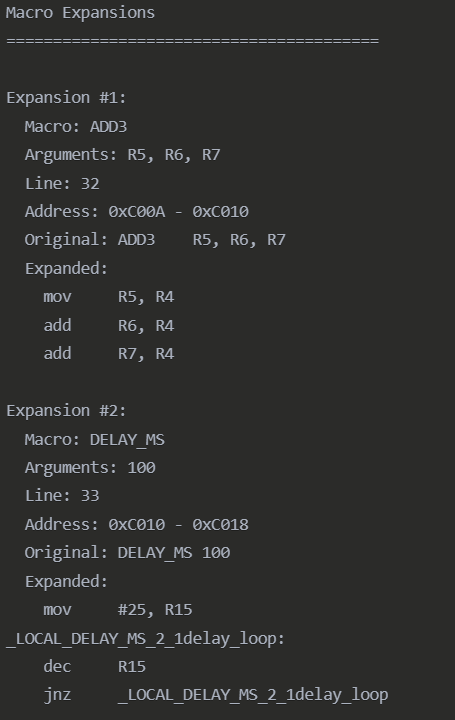
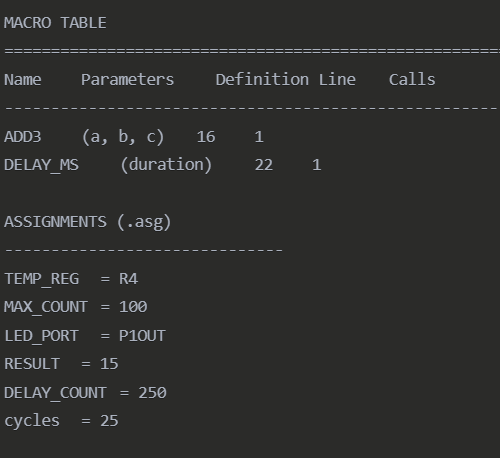
* **.macro/.endm** direktifleri ile makro tanımı
* **.asg** direktifi ile sembol ataması
* **.eval** direktifi ile matematiksel değerlendirme
* **.if/.else/.endif** koşullu derleme
* **.loop/.endloop** döngü yapıları
* **.mlist/.mnolist** makro listeleme kontrolü

### 2.2 Giriş ve Çıkış Örneği

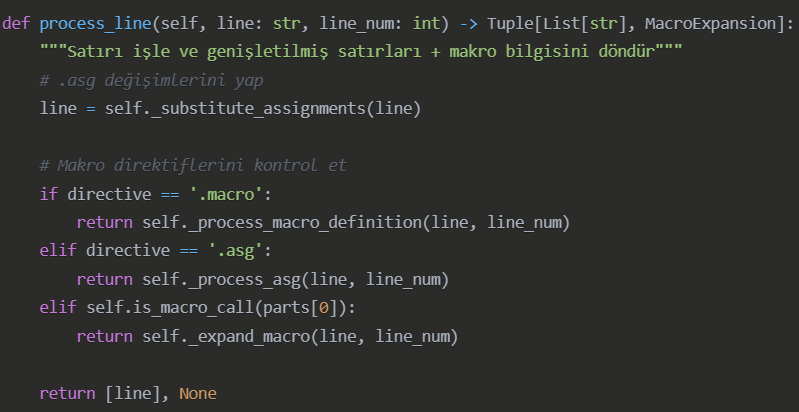
**Giriş Dosyası (test\_macro.asm):**



**Çıkış Dosyası (.mac Makro Tablosu):**



**2.3 Makro İşleme Algoritması**



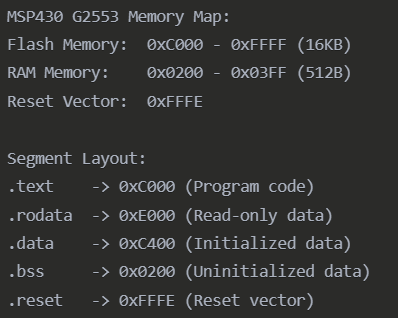
## 3. ELF Linker (msp430\_linker.py)

### 3.1 Teknik Özellikler

ELF linker modülü, çoklu object dosyasını birleştirme ve memory layout organizasyonu sağlamaktadır:

* **Symbol Resolution**: Global ve local symbol çözümlemesi
* **Section Merging**: .text, .data, .bss, .reset section'larının birleştirilmesi
* **Relocation Processing**: R\_MSP430\_16 tipi relocations
* **Memory Layout**: MSP430 G2553 memory map'ine uygun adres ataması
* **Conflict Resolution**: Çoklu tanımlar için öncelik sistemi

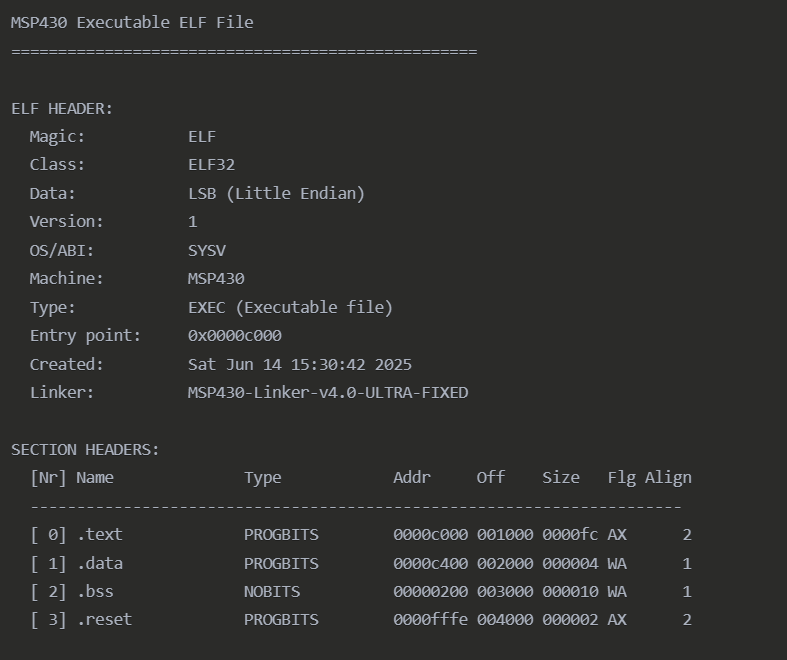
**3.2 Memory Layout**

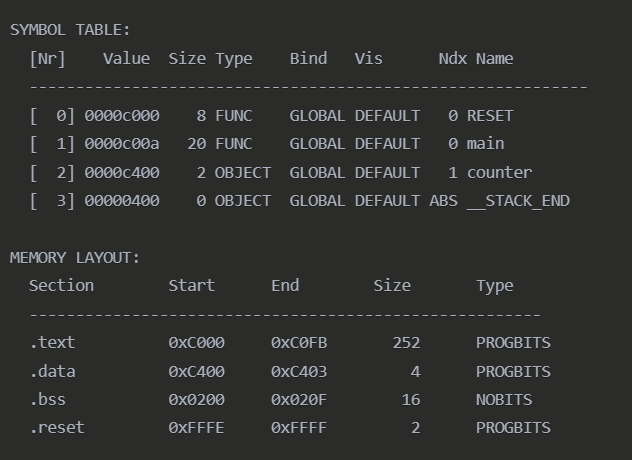
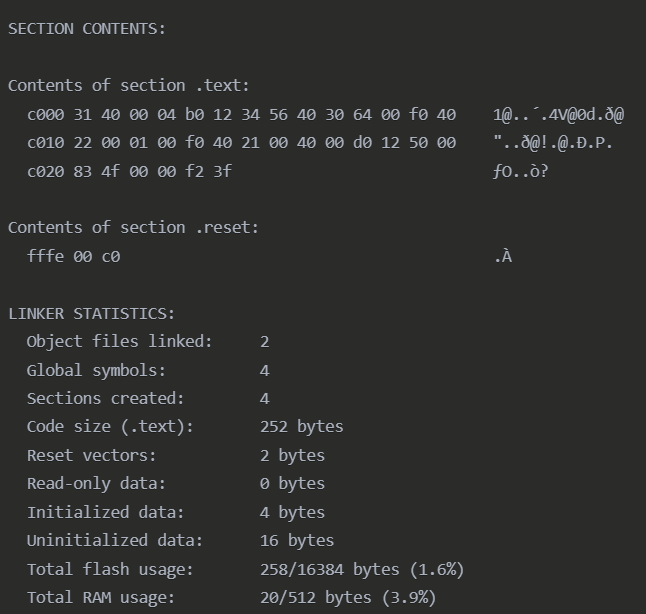


**3.3 Giriş ve Çıkış Örneği**

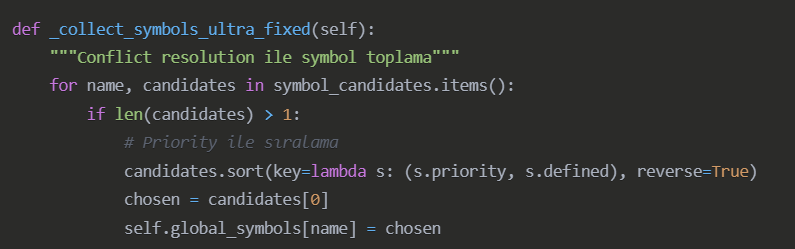
**Komut:** python msp430\_linker.py program.elf main.obj startup.obj

**Çıkış Dosyası (program.elf):**





**3.4 Symbol Resolution Algoritması**



## 4. Flash Loader (msp430\_loader.py)

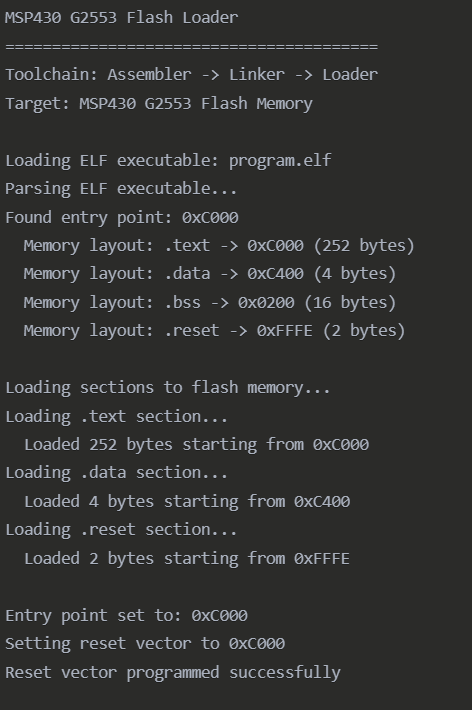
### 4.1 Teknik Özellikler

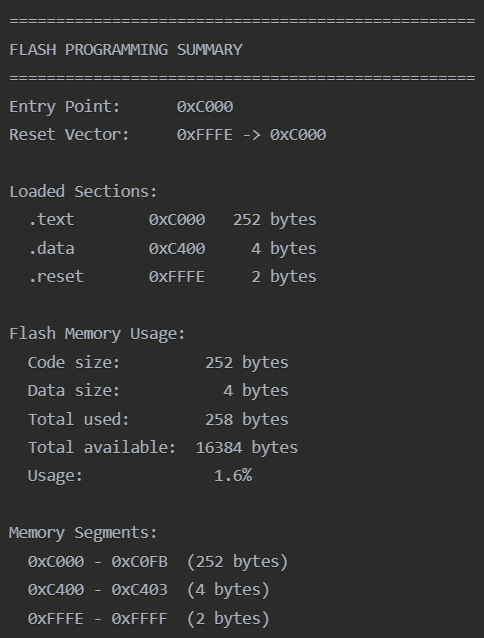
Flash loader modülü, ELF executable dosyalarını MSP430 flash memory'sine programlama için hazırlamaktadır:

* **ELF Parsing**: Linker çıktısı ELF dosyalarının parse edilmesi
* **Memory Simulation**: MSP430 G2553 flash memory simülasyonu
* **Intel HEX Generation**: Programlama araçları için standart format
* **Memory Dump**: Debug amaçlı memory görüntüleme

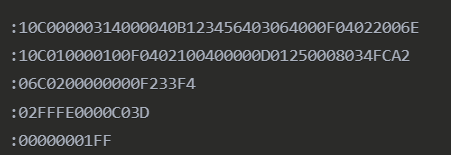
### 4.2 Giriş ve Çıkış Örneği

**Komut:** python msp430\_loader.py program.elf

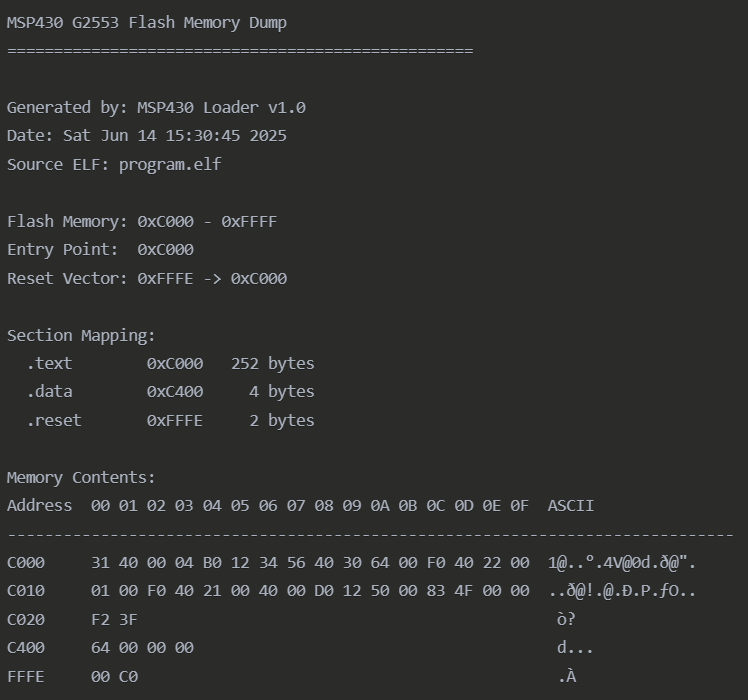
**Konsol Çıktısı:**



**Intel HEX Çıktısı (program.hex):**



**Memory Dump Çıktısı (program.dump):**



## 5. Assembler ELF Object Format Geliştirmeleri

### 5.1 Yeni ELF Object Writer

elf\_object\_writer.py modülü ile assembler çıktısında ELF uyumluluğu sağlanmıştır:

**Çıkış Örneği (.obj dosyası):**

