

1 The stack pointer is a register(s) that holds the memory location of the next item on the stack. The stack operates as “last in first out” and can be interacted with using the push or pop commands. Push, places an item onto the top of the stack. Pop, removes the top of the stack for use. The stack is initialized by loading the stack pointer with the starting memory address of the stack, usually low(RAMEND).

Something like

```
Ldi rd, RAMEND
```

```
Out SPL, rd
```

2LPM: load program memory. This command loads one byte, pointed to by the z register, from data memory into a designated register. To use lpm, one must first load the z register with the location in memory of the desired data, and then invoke lpm with the destination register.

Something like

```
Ldi ZH, ADDRESS
```

```
Ldi ZL, ADDRESSS
```

```
Lpm rd, Z
```

3 The definition file provides a library of pre-defined values for a specific hardware. This allows developers to use defined expressions or variables in the place of hardware values. For example I/O register mapping would be specific to hardware, so the memory location of port b would change based on the hardware. Definition files provide the values so that the expression DDRB can be used instead of the 0x17 which is specific to this board. Another example would be values related to timers.