

# Programming the OTP Registers in JN516X Devices

---

**NOTE :** The OTP API is released to customers on a "need to know" basis rather than as part of the SDK to prevent unwary customers from trashing their devices.

The API functions are included in the Hardware API library file, but the headers are not released. These are attached in the zip and should be copied into the Component\HardwareAPI\Include Folder of your SDK.

There are 8 functions, 4 for writing and 4 for reading the values back. All write functions return FALSE if the field has already been written, and all functions return FALSE if the input parameters are out of range.

```
PUBLIC bool_t bAHI_WriteCustomerMACID(  
    uint64 u64CustomerMACId);
```

Writes the supplied MAC address into the "customer MAC address" field.

```
PUBLIC bool_t bAHI_WriteCustomerAESKey(  
    uint32 au32AESkey[4]);
```

Writes the supplied AES key into the security key field.

```
PUBLIC bool_t bAHI_WriteUserData(  
    uint8 u8UserFieldIndex,  
    uint32 au32userData[4]);
```

Writes into one bank of the user data area. There are 3 user data banks of 128 bits each, passed in u8UserFieldIndex; hence it can take values 0 to 2.

```
PUBLIC bool_t bAHI_WriteCustomerSettings(  
    bool_t bJTAGdisable,  
    uint8 u8VBOthreshold,
```

```
uint8  u8CRP,  
bool_t bEncryptedExternalFlash,  
bool_t bDisableLoadFromExternalFlash);
```

Writes several miscellaneous system configuration fields. Each bool\_t can be either TRUE or FALSE (TRUE to disable JTAG, disable load from external flash or when using encrypted external flash). The VBO threshold and CRP values have values defined in the header file.

```
PUBLIC bool_t bAHI_ReadMACID(  
    bool_t bReadCustomerMACID,  
    uint64 *pu64CustomerMACId);
```

Reads back the MAC address. Passing TRUE in bReadCustomerMACID causes the customer MAC address to be read, otherwise the default MAC address is read.

```
PUBLIC bool_t bAHI_ReadCustomerAESKey(  
    uint32 au32AESkey[4]);
```

Reads back the security key.

```
PUBLIC bool_t bAHI_ReadUserData(  
    uint8 u8UserFieldIndex,  
    uint32 au32userData[4]);
```

Reads back user data.

```
PUBLIC bool_t bAHI_ReadCustomerSettings(  
    bool_t *pbJTAGdisable,  
    uint8 *pu8VBOthreshold,  
    uint8 *pu8CRP,  
    bool_t *pbEncryptedExternalFlash,  
    bool_t *pbDisableLoadFromExternalFlash);
```

Reads back the miscellaneous settings.